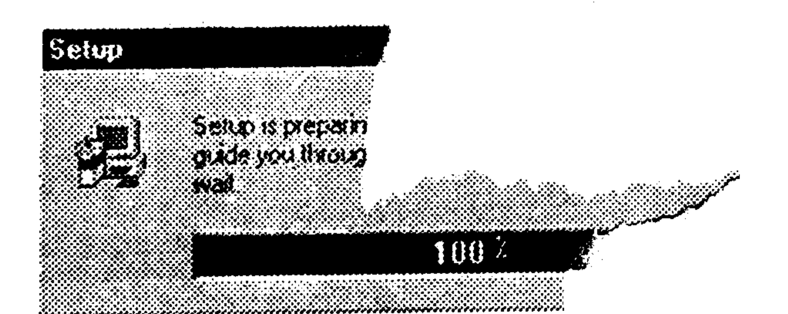
MySQL



بسم الله الرحمن الرحيم

((نرفع درجات من نشاء وفوق كل ذي علم عليم))

صدق الله العظيم

سورة يوسف (آية ٧٦)

MR

تمهيد

مع التطورات الحديثة التي يشهدها العالم وبالتحديد في مجال الحاسبات الآلية وما يصاحبها من برمجيات أيضاً متطورة ومختلفة ومتخصصة فى معالجة قطاعات معينة من ميادين الحياة، ظهرت في بداية التسعينيات برمجة ذات طبيعة خاصة مرتبطة ببرمجة شبكات الويب Web، وبالتالي ظهرت لغات جديدة وبرامج جاهزة أيضاً جديدة ذات علاقة بهذه النوعية من البرمجة وتطورت برامج قديمة لتتماشى مع برمجة الـ WEB، منها لغات:

ASP, ASP.Net, PHP, XML, Perl, JAVA ,
Java Script, VB Script ...

والبرامج الجاهزة منها برامج تصميم الصفحات Fornt Page و DreamWeaver و Flash وغيرها وبرامج الاتصال بالمواقع لرفع أو تحميل الصفحات مثل برامج FTP وغيرها من البرامج المتعددة، وأيضاً برامج قواعد بيانات منها

MySQL , PostgreSQL , SQL Server , ORACLE .

ولذا يجب على المهتمين بعلوم برمجيات الحاسب أن يسايروا التطورات الحديثة المرتبطة بعلوم تخطيط البرامج المصاحبة للشبكات والانترنت، حيث يمكن أن يتوافر لمخططة البرامج أساليب مختلفة لتطوير برمجياتهم وخصوصاً المرتبطة بالانترنت، والتي تتمتع بالقوة والمرونة وبالطبع تقديم الخدمات المختلفة للمستخدمين، ولذا فهذا الكتاب يمكن أن يخاطب الأفراد الذين تعلموا من قبل تخطيط البرامج على الأقل من خلال لغات برمجة وأيضا يستحسن أن يكون لديهم خبرة فى أساسيات البرمجة باستخدام لغة HTML وهى إحدى اللغات المرتبطة بتصميم صفحات الانترنت والتي لها علاقة بقواعد بيانات الانترنت.

ومن خلال هذا الكتاب سوف يتم التعرف على أحدى أدوات تطـوير الويـب Web والخاصة بقواعد بيانات الويب، وذلك من خلال برنامج MySQL والتي تعتبر إحدى قواعد البيانات العلائقية Relational Databases والتي يمكـن أن تتعامل مع عدد من المستخدمين فى نفس الوقت.

وسيتناول هذا الكتاب بعض الفصول المرتبطة بالمادة العلمية التالية:

- مدخل لقواعد البيانات DATABASES.
- مدخل نظرى للغة SQL.
- مدخل نظرى للتعرف على نظام (البرنامج الجاهز) MYSQL وأهميته.
- مراحل مختلفة للتشغيل العملى سيتم التعامل معها من خلال MySQL:
  - مرحلة تصميم قاعدة البيانات والجداول.
  - مرحلة الإضافة لبيانات القاعدة والجداول.
  - مرحلة التعديل لبيانات القاعدة والجداول.
  - مرحلة التشغيل والمعالجة لبيانات القاعدة والجداول.

وبالتالي يمكن القول أنه يجب على المهتمين ببرمجة الويب أن يتعلمـوا لغات وبرامج جاهزة مرتبطة ببرمجة الويب والتي سيتيح هذا الكتاب التعرف على أساسيات إحدى هذه البرامج وهى المرتبطـة بقواعـد بيانـات الويـب وهى MySQL.

الفصل الأول

مدخل نظرى لقواعد البيانات
Introduction to Databases

يتناول هذا الفصل المحتويات الآتية :

- فكرة عامة عن قواعد البيانات ومميزاتها
- فكرة عامة عن بعض نظم قواعد البيانات الجاهزة

١ - ١ فكرة عامة عن قواعد البيانات ومميزاتها DATA BASE S

تعتبر برامج نظم قواعد البيانات من أشهر البرامج المستخدمة على الحاسبات الآلية لما لها من قدرة على التعامل مع حل مشاكل وحاجات العملاء ، وبالتالي المساعدة في تطوير وتيسير أعمالهم . وأيضا تعد هذه النوعية من البرامج من أوسع برامج الكمبيوتر الشخصي حيث يمكن استخدامها في مختلف الميادين أو المجالات التجارية والصناعية ، ومنها على سبيل المثال لا الحصر المجالات المالية والمحاسبية والتسويق - المبيعات والمشتريات وشئون الأفراد والتخزين ....الخ .

وبالطبع هذا العلم فى الحقيقة يمكن تنفيذه يدويا أو بطريقة آلية ، حيث يمكن استخدام الدفاتر والسجلات لعمل وتشغيل قواعد بيانات يدوية ، ويمكن استخدام أنظمة آلية مثل قواعد البيانات DBASE , FOX , ACCESS لمعالجة البيانات بطريقة آلية ، وبالطبع تختلف الطريقتين فيما بينهما اختلافاً كبيرا من حيث التشغيل والنتائج .

وبصفة عامة يمكن تعريف قاعدة البيانات على أنها: " تجميع لكمية من البيانات المتصلة بموضوع ما أو غرض ما مرتبط بظاهرة ما في حياتنا، ثم عرض هذه البيانات بأسلوب أو بطريقة ما بعد إجراء عمليات الفرز والترتيب والعمليات الحسابية والمنطقية وعمليات المقارنة...الخ، وذلك بهدف تسهيل الاستفادة منها".

ويمكننا التعرف على مثال بسيط وواضح عن قواعد البيانات وهو بيانات المشتركين فى دليل التليفون، حيث يتم تخزين البيانات المختلفة عن المشتركين مثل رقم التليفون واسم المشترك وعنوانه وطبيعة عملة ...الخ..، وتظهر أو تتحقق الفائدة من هذه القاعدة عندما تبحث عن بيانات مشترك ما وذلك بإدخال رقم التليفون ويظهر لك جميع البيانات الأخرى التي تحتاجها.

MR

مثال أخر لمعالجة بيانات الطلاب بكلية التربية النوعية بدمياط لإظهار نتيجة الطلاب بطريقة آلية، ويلزم لذلك وجود ملفات ليسجل بها بيانات الطلاب تسمى ملفات بيانات تختلف طبيعتها وحدودها من نظام لأخر وأيضا يلزم وجود ملفات برامج تحتوى على تعليمات وأوامر لمعالجة محتويات أو عناصر البيانات الموجودة فى ملفات البيانات. ومن هنا يمكن القول بأن ملفات البيانات للقاعدة وملفات البرامج لمعالجة هذه البيانات هى نظام متكامل لإدارة قاعدة البيانات.

ومن خلال التجهيز الآلي للبيانات وتحقيق الأهداف المرجوة منه، فإنه يلزم وجود أنظمة معينة للإدارة وتنظيم البيانات المخزنة على الحاسب، وهذه الأنظمة هي ما يطلق عليها (نظم إدارة قواعد البيانات Database Management Systems) ، ويمكن تعريفها على أنها مجموعة متنوعة من البرامج الجاهزة التي تؤدى إلى تحقيق مجموعة من الأهداف المؤثرة على الأنشطة الرئيسية في مجالات تطبيقات معالجة البيانات الآلية والمطلوب منها تنفيذ جميع الوظائف المراده من قاعدة البيانات، ويتكون نظام إدارة قاعدة البيانات من مجموعة من الملفات بالإضافة إلى البرنامج أو البرامج التي تلحق بالقاعدة لحل مشكلة القاعدة أو لتحويل معالجة المشكلة من المعالجة اليدوية إلى المعالجة الآلية التي تستخدم قدرات الحاسب الآلي.

١-١-١ أنواع قواعد البيانات:-

بعد ظهور هذا العلم في أوائل الستينيات الميلادية بدأ يتطور وظهر له عدة أنواع وأشهر هذه الأنواع ما يلي:

١ - قواعد البيانات الهرمية Hierarchy Databases

٢ - قواعد البيانات الشبكية Network Databases

٣ - قواعد البيانات الموجهة Oriented Databases

٤ - قواعد البيانات العلائقية Relational Databases

بالنسبة للنوع الأول والثاني والثالث ففي الغالب يتم تشغيلهما على الأجهزة الكبيرة والأجهزة المتوسطة وبالتالي فلها مميزات وقدرات كبيرة وتحتاج لمستخدمين ذو قدرات عالية حيث تحتاج إلى لغات برمجة راقية وبالطبع صعبة التعلم بالإضافة إلى الحاجة إلى ذاكرات ذات أحجام كبيرة، مع العلم أيضاً أن النوع الرابع يمكن أن يعمل على الأجهزة الكبيرة مع اختلاف أنظمة قواعد البيانات، ويمكن القول أنا قواعد البيانات العلائقية فهي مخصصة بدرجة عالية للعمل على الأجهزة الصغيرة لاختلاف قدراتها وإمكانياتها عن الأنواع السابقة، مع توافر أنظمة جاهزة مختلفة لتطبيقها.

على سبيل المثال قواعد البيانات العلائقية التي توافر علي أجهزةIBM-PC أو الأجهزة المتوافقة معها تختلف بالطبع عن قواعد البيانات التي تعمل على أنواع أخرى من الأجهزة، ويعتبر برنامج قواعد البيانات ACCESS المستخدم من خلال بيئة برنامج تشغيل النوافذ WINDOWS ، من أسهل الأنواع في التعامل وبرمجتها ، بالإضافة إلى أنها لا تحتاج إلى ذاكرة كبيرة أو أوساط تخزين كبيرة ، ومن أشهر قواعد البيانات المستخدمة على الحاسبات الشخصية ، بالإضافة إلى أنواع أخرى من نظم قواعد البيانات ذات الطبيعة المختلفة عن السابقة والتي تعمل على نفس الأجهزة.

ولكن هذا الكتاب سيتعامل مع نوعية أخرى من الأنظمة أو البرامج مخصصة للعمل على أنواع مختلفة من الأجهزة نظراً لقدرتها ومميزاتها نتيجة استخدامها مع الانترنت أو الاستفادة منها في قواعد بيانات شبكة المعلومات الدولية والمسماة MySQL ، والتي سيتم التعرف على أساسياتها من خلال هذا الكتاب كإحدى الأنظمة الجاهزة المتقدمة.

١-١-٢ كيفية تمثيل البيانات في قواعد البيانات العلائقية:

يتم تمثيل البيانات في هذا النوع في صورة جداول (ملفات مستقلة) ، حيث يتكون الجدول من مجموعة من الصفوف ومجموعة من الأعمدة ويقابل ذلك ملفات البيانات FILES ويقابل الصفوف سجلات RECORDS ويقابل الأعمدة حقول FIELDS .

كل جدول من هذه الجداول مرتبط مع الجداول الأخرى بعلاقة محددة مسبقاً من ضمن بيانات الجدول، وبالتالي تعتبر أبسط صور قواعد البيانات العلائقية هي التي تستخدم ملفاً واحداً يحتوى على البيانات. وهذا الملف يحتوى على طبيعة واحدة من البيانات وليكن البيانات الشخصية للطالب مثل رقم البطاقة ، اسم الطالب ، النوع ، العنوان ...الخ. وبالتالي يمكن أن يكون هناك ملفات أو جداول أخرى ترتبط بعلاقة مع بعضهما البعض وليكن من خلال حقل مفتاح معين يتم تكراره هو فقط في الملفات الأخرى.

وبالتالي يمكن تشغيل البيانات في الجداول المختلفة وإخراج النتائج المتنوعة من الجداول المختلفة بناءاً على الحقل المفتاح أو الحقل المكرر في الجداول.

ميزة تعدد الجداول الخاصة بهذه النوعية من قواعد البيانات تتيح للمستخدمين سهولة التعامل مع كل ملف على حدة بخلاف أن يكون هناك ملف واحد يحتوى على بيانات كثيرة ومختلفة ذات طبيعة أيضاً متنوعة، ولذا التعامل مع جدول واحد يحتوى على عناصر بيانات ذات طبيعة واحدة يؤدى إلى توافر كثير من المزايا وتقوية عمليات التشغيل المختلفة على البيانات في مراحلها المختلفة.

وبالطبع وجود هذا الحقل المفتاح في الجداول المختلفة يوفر كثير من الوقت والمساحة التخزينية، حيث لا يتم تكرار البيانات الثابتة مثل اسم الطالب أو

النوع أو العنوان فى الجداول الأخرى ويتم التعامل مع بيانات الجداول المختلفة من خلال العلاقة الموجودة في هذه الجداول وهو الحقل المكرر.

هذا التمثيل للبيانات يشابه إلى حد ما تمثيل البيانات في جداول البيانات الخاصة بأنظمة الجداول الإلكترونية المسماة SPREED SHEETS ومنها برامج :

LOTUS 123 , QUATTRO , EXCEL ...

مع إخلاف طرق التخزين والتشغيل وبالطبع الأغراض والوظائف.

١-١-٣ مزايا قواعد البيانات ذات العلاقة (العلائقية) :

من خلال الشرح السابق يمكن إيجاز بعض المزايا التى تحققها قواعد البيانات العلائقية فيما يلي:

١ - زيادة كفاءة إدارة البيانات :

بالطبع في حالة وجود عناصر بيانات (حقول) قليلة فى الملفات (الجداول) فإن ذلك سيساعد على سهولة التعامل مع البيانات ، لأن وجود كثير من الحقول سيؤدى إلى صعوبة العمل مع البيانات ، وهذا يتضح من أن قواعد البيانات العلائقية تسهل التعامل مع إدارة البيانات نتيجة وجود بيانات محددة وغير مكررة فى الجداول ، وبالتالي يسهل إدارة البيانات وزيادة كفاءتها .

٢ - زيادة كفاءة تنظيم البيانات :

أيضاً تتيح قواعد البيانات العلائقية إنشاء علاقة بين بيانات الجداول المختلفة للقاعدة وبالتالي يمكن الاستفادة من العناصر المختلفة للجداول وإظهار النتائج أو البيانات فى صور أو أشكال مختلفة ، وبالتالي يمكن زيادة كفاءة تنظيم البيانات .

MR

٣ - اختصار البيانات :

أيضاً تتيح قاعدة البيانات العلائقية عدم تكرار بعض عناصر البيانات في الجداول نتيجة استخدام حقل واحد يتم ربطة بجداول قاعدة البيانات ، وبالتالي يمكن التعامل مع الجداول المختلفة من خلاله ولا يلزم وجود بيانات أساسية أو استرشادية في الجداول الأخرى لمعرفة أصحابها ، مثل عدم تكرار البيانات الأساسية والتي يمكن أن تكوم موجودة في الجدول أو الملف الأول من القاعدة مثل بيانات (اسم الطالب - النوع - تاريخ الميلاد ...الخ) وبالتالي يمكن توفير كثير من وقت التشغيل والمساحة التخزينية على أوساط التخزين وتقليل فى كثير من التكاليف مثل تكاليف الاستخدام .

٤ - مرونة التشغيل :

وجود جداول كثيرة وعناصرها قليلة بالطبع يؤدى ذلك إلى مرونة كبيرة فى التشغيل للبيانات فى الجداول المختلفة أو في إيجاد علاقة بين عناصر هذه الجداول واستخراج النتائج المختلفة بسهولة ويسر منها ، حيث يختص كل جدول كما نعرف بطبيعة واحدة من البيانات . وبالتالي يسهل الربط بين الجداول المختلفة نتيجة وجود حقل العلاقة أو الحقل الرابط بين الجداول .

بالإضافة إلى مزايا أخرى سيتم التعرف عليها والمرتبطة بمزايا MySQL فيما بعد.

١ - ٢ فكرة عامة عن بعض نظم قواعد البيانات الجاهزة ومميزاتها

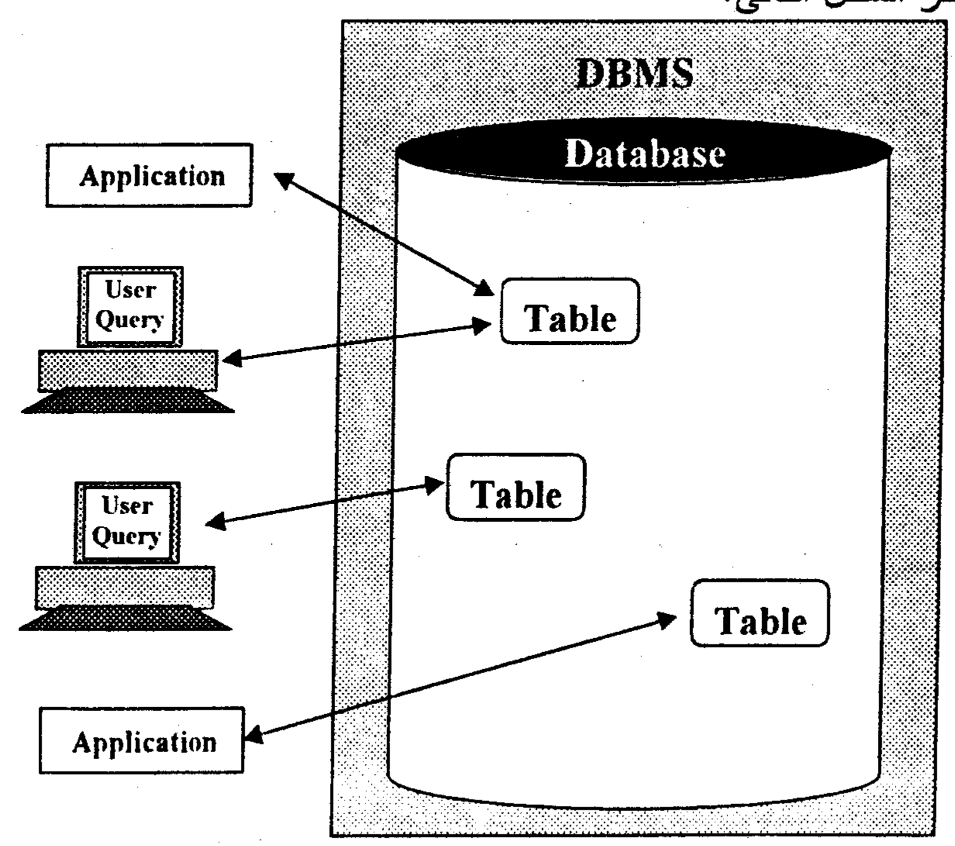
فيما يلي يمكن التعرف على بعض أنظمة قواعد البيانات العلائقية المنتشرة على أنواع مختلفة من الأجهزة وسنتعرض أيضا إلى معلومات كافية لفهم مبدأ و قدرة قواعد البيانات العلائقية فى تخزين و استرجاع البيانات، وسنتعرف على مبادئ ثلاثة أنظمة RDBMS تعمل على الحاسبات الكبيرة

(mainframes) و هي INGERS , DB2 , ORACLE ، و بالطبع هنـــاك أنظمة أخرى بالإضافة إلى الأنظمة الثلاثة هذه تعمل أيضـــا علــى الحاســبات الشخصية (Personal Computer) إلا أننا لن نتطرق إليها، وفيما يلــي نبــذة مختصرة نعرضها كما يلي:

١ – ٢ – ١ التعرف على نظام إدارة قواعد البيانات DBMS:

يشير المصطلح " نظام إدارة قواعد البيانــات " أو DBMS – اختصــارا للعبارة Database Management System – إلى نظام إدارة وإنشاء ملفــات تعتمد وترتكز على السجلات بطريقة الكترونية.

انظر الشكل التالي:



MR

تأتى أنظمة DBMS بأصناف عديدة و بمزايا مختلفة إلا أنها بشكل عام تسعى لتحقيق ثلاثة أشياء رئيسية هي:

١. دمج البيانات (data consolidation)
٢. المشاركة على البيانات (data sharing)
٣. حماية البيانات (data protection)

حيث يمكن أن يشير المصطلح الأول " **دمج البيانات** " إلى إمكانية ضم أو توحيد ملفات البيانات المختلفة والمنفصلة فى بنية (هيكل - تركيبة) مركزية وتخزين البيانات فى صيغة خالية من الزيادة أو الفائض (No redundant). ينشأ الفائض فى بنية البيانات عندما تخزن نفس عناصر البيانات فى موقعين أو أكثر.

فمثلا قد نجد *عنوان الموظف* مخزنا ليس فقط فى ملف الرواتب (payroll) و إنما أيضا فى ملف الإدارة أو ملف القسم التابع له وملف المشروع وملف الموظف نفسه، وعندها نكون أمام بنية غير مركزية تحوى معلومات زائدة. ولذا إذا كان النظام يتصف بالمثالية ومتكاملا (خاليا من الفائض) حيث يمكن أن يحتوى على عنوان الموظف فى موقع وحيد.

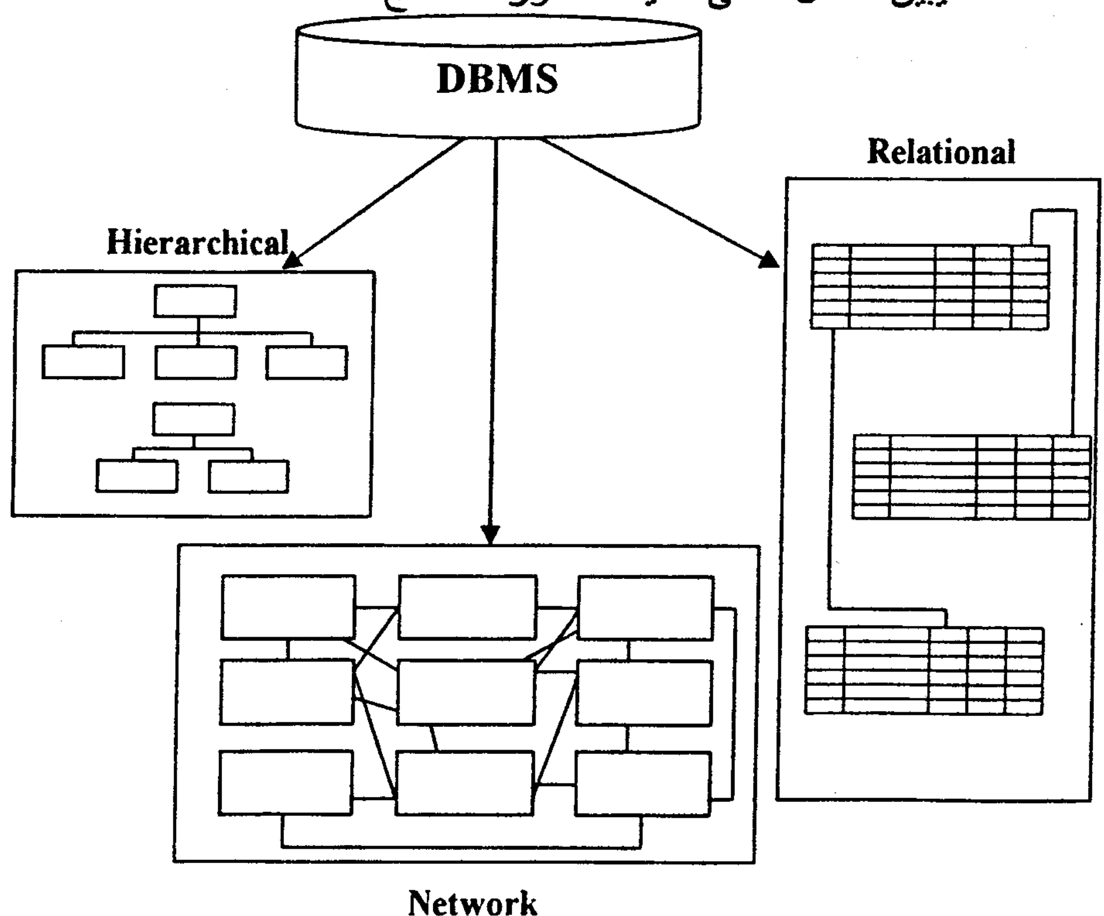
أما مصطلح " **المشاركة على البيانات** " فهو يشير إلى قدرة النظام على السماح لعدة مستخدمين بالوصول إلى أجزاء مستقلة من البيانات ضمن القاعدة فى نفس الوقت.

ويمكن أن يشير المصطلح " **حماية البيانات** " إلى قدرة نظم DBMS على المحافظة على سلامة البيانات فى حالة حدوث مشاكل أو حوادث مفاجأة خلال المعالجة (عطل البرنامج أو توقفه...الخ) ولذا يجب على DBMS أن يملك القدرة على إعادة البيانات إلى حالتها السابقة قبل التعديل غير الكامل عليها أو قبل حدوث الخطأ فيها (تسمى هذه العملية أحيانا بالتراجع UNDO إلى الحالة السابقة).

عندئذ فنظم DBMS تتحكم فى وتنظم البيانات فى بنيـــة تـــدعى نمـــوذج البيانات(Data Model)، حيث يمكن النظر إليها على أنها البنيـــة التحتيـــة أو الهيكلية لتنظيم البيانات، أو بمعنى أخر كيفية تقديم البيانات إلى المستخدم، حيـــث يوجد نماذج مختلفة للبيانات تصنف طبقاً لأنواع نظم قواعـــد البيانـــات الســـابق التعرف عليها ومنها ما يلي:

١. النموذج الهرمي Hierarchical
٢. النموذج الشبكي Network
٣. النموذج العلائقى Relational

يبين الشكل التالى تمثيلا مصورا للنماذج الثلاثة:



MR

* و ألان ما هو الفرق بينهم ؟

يستخدم النموذجان الهرمي و الشبكي روابط أو مؤشرات لوصل السجلات ببعضها فى النظام ، و تدعى أمثال هذه الأنظمة بالأنظمة الإستاتيكية أو المتراصة لان السجلات فيها مربوطة ببغضها بشكل فيزيائي أو مادي من خلال تعريفها، تتميز هذه الأنظمة بأنها معقدة العمل و صعبة التعديل ، إلا أن سرعة الوصول فيها تغطى عيوبها.

ولكن فى الأنظمة العلائقية فالربط بين السجلات أو إيجاد العلاقة لا يجرى فيزيائيا عن طريق المؤشرات وإنما عن طريق الأسماء الحقيقية للحقول، كحقل رقم الموظف Emp_nu أو رقم البطاقة ID أو حقل اسم الموظف Emp_name ، ... الخ. فالسجلات فى هذا النظام قابلة للعنونة بمحتوياتها بحيث يجرى الوصول إليها بمطابقة قيم البيانات المخزنة مع بعضها البعض.

** ماهية النموذج العلائقى "DBMS" What is a Relational Model:

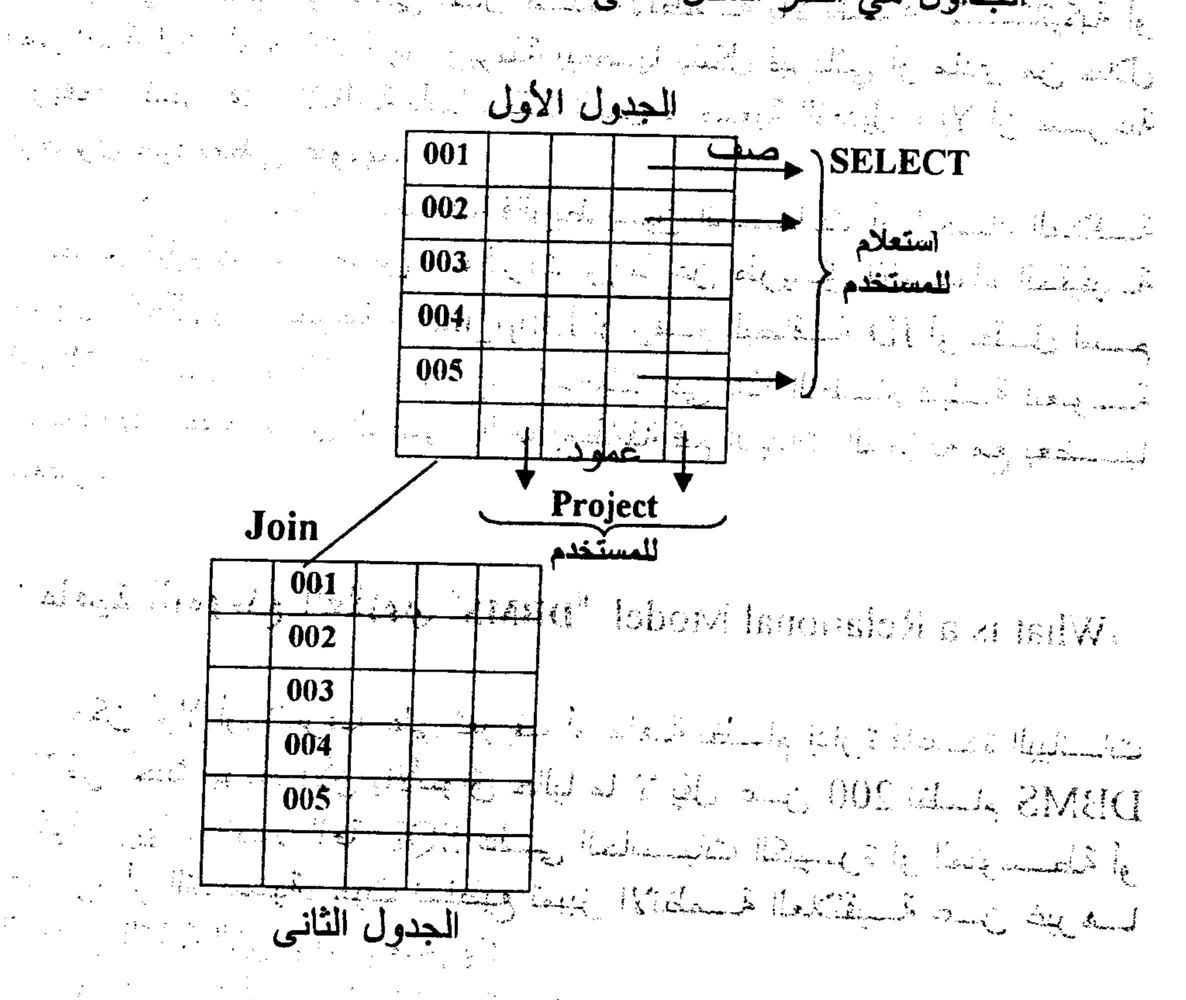
يمكن أولا أن نتعرف على تعريف أو ماهية نظام إدارة قاعدة البيانات العلائقى، حيث يوجد الآن بالأسواق حاليا ما لا يقل عن 200 نظام DBMS نصفها تقريبا يستخدم اللغة SQL على الحاسبات الكبيرة أو المتوسطة أو الصغيرة أو الشخصية، حيث نستطيع تمييز الأنظمة العلائقية عن غيرها بالمعايير التالية التى توضح ماهيتها:

١. تسمح الأنظمة العلائقية بتقدم البيانات إلى مستخدمي القاعدة أو النظام على شكل مجموعة من الجداول البسيطة ثنائية البعد (حيث تتكون من مجموعة من الأسطر و الأعمدة).
٢. تسمح الأنظمة العلائقية للمستخدمين بتوليد أو إنشاء جداول جديدة بالاعتماد على الجداول الموجودة حيث يمكن لهذه الجداول الجديدة

MR

المولدة أن تكون بناءً لبيانات مؤقتة (Temporary)، وفى الغالب تكون نتيجة استفسار (Queries) تستخدم كتقارير قد تكون دائمة (Permanent).

٣. تسمح الأنظمة العلائقية بإجراء ثلاث عمليات علائقية مميزة على الجداول هي انظر الشكل التالى :



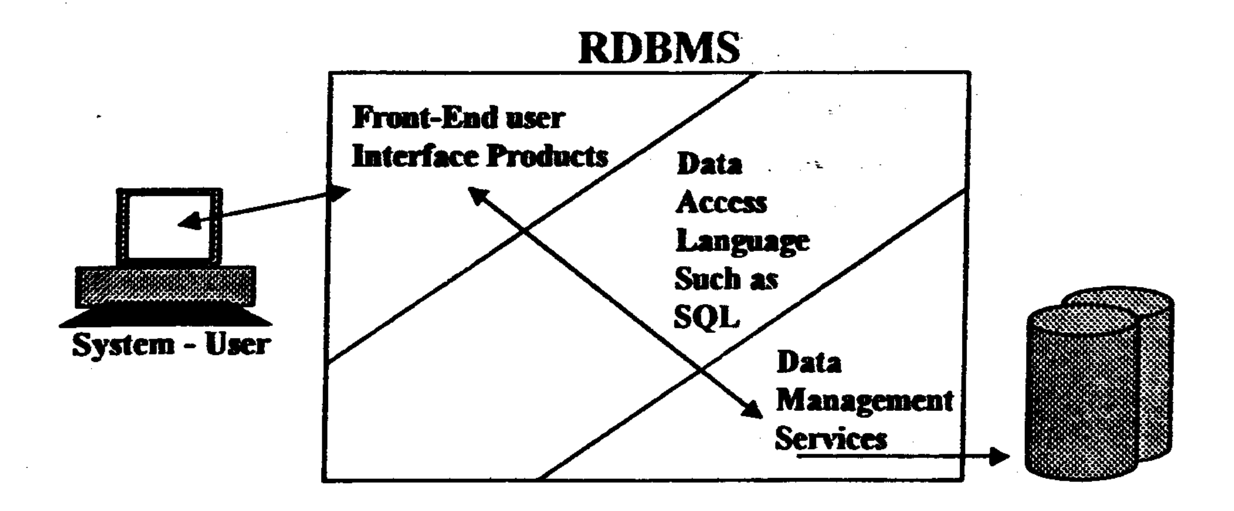
- SELECT : التحكم فى الوصول (ACCESS) إلى سجلات أو أسطر معينة فى الجدول بسهولة.

- PROJECT: التحكم فى الوصول إلى عناصر معينة أو حقول أو أعمدة معينة فى الجدول بشكل مرن.
- JOIN : التحكم فى دمج بيانات أكثر من جدول واحد معا.

١ - ٢ - ٢ العناصر الأساسية لقواعد البيانات العلائقية RDBMS :

العناصر الأساسية لهذه الأنظمة RDBMS تتمثل فى:

انظر الشكل التالى :



١ - واجهة المستخدم الأمامية (Front-End Interface) :

حيث تمثل مجموعة البرامج التى تتعامل وتتفاعل مع مستخدم النظام ، كمولد التقرير Report Generator و أدوات تطوير Tools Development

واجهة التطبيق Interface Applicant و ...... الخ ، و لهذه المجموعة فائـدة كبيرة إذا احتوت على أدوات تطوير التطبيقـات ، وأدوات داعمـة أو مسـاندة للقرار (Decision) .

٢ - الخدمات الخلفية لإدارة البيانات

(Back- End Data Management Services):

حيث تمثل أو تكون أو تشكل القلب المحرك للنظام، فهي تزودنـا بإمكانيـة الوصول للبيانات و المحافظة على سلامتها، وذلك باستخدام آليات التحكم بها.

٣ لغة الوصول إلى البيانات معطيات (Data Access Language):

حيث تؤمن اتصال طلبات مستخدم النظام مع خدمات إدارة البيانات، وفـي معظم الأحيان هذه اللغة هي SQL، والتي سنفرد لها فصلا مسـتقلاً مـن هـذا الكتاب للتعرف على ماهيتها وأساسيات التعامل معها.

وتعتمد أو تركز بعض أنظمة RDBMS مثـل ORACLE و INGERS بشكل مكثف على خدمات ووظائف DBMS فهي تزودنا بمجموعة وافرة مـن برامج الواجهة الأمامية، أما قوة DB2 و SQL/DS فهي تكمـن فـي آليـات الحفاظ على تجانس وتماثل البيانات و التحكم المعقد بالبيانات.

١ – ٢ – ٣ نظرة سريعة على بعض أنظمة قواعد البيانات العلائقية RDBMS :

ومنها : DB2, ORACLE, INGERS

أولاً : نظام قاعدة البيانات DB2 :

يمثل DB2 اختصارا للعبارة DATA BASE 2 أي نظـام IBM لإدارة قواعد البيانات العلائقية للحاسبات الكبيرة التى تعمل تحت أنظمة التشغيل MVS/SP, MVS/XA, MVS/ESA وغيرها.
حيث أطلقت IBM الإصدار الأول من DB2 فى عام 1983 و بقى لمدة عقد كامل أكثر نظام يتحدث عنه الناس، حيث تكمن قوة DB2 فى قدرته الفعالة والمؤثرة فى إدارة البيانات وسرعته فى معالجة الحركات التى بلغت اكسر من ٤٣٠ حركة فى الثانية فى تطبيق دائن / مدين نموذجي عند قياسها ببرامج تحليل الأداء.

- بيئة DB2 :

يعمل نظام قاعدة البيانات DB2 تحت تحكم وإشراف احد مدراء الحركة حيث تمثل الحركة أية وحدة كاملة من أوامر التشغيل أو المعالجة، وعلى هذا فإن مدير الحركة هو برنامج مسئول عن التنفيذ الناجح لمجموعة أعمال تؤديها الحركة داخل النظام.

* TSO : خيار المشاركة الزمنية (TIME SHARING OPTION)
* IMS DC : اتصالات بيانات أنظمة إدارة المعلومات

Information Manager System Data Communication.

MR



* CICS : نظام التحكم بمعلومات الزبون
CUSTOMER INFORMATION CONTROL SYSTEM.

– يزود مدير الحركة النظام DB2 بخدمات أساسية تتضمن ما يلى :

١ – اتصالات البيانات (DATA COMMUNICATION) : وتعني القدرة على تأمين اتصال المستخدمين من نهايات طرفية بعيدة (remote terminal) .

٢ – تجانس البيانات (DATA INTEGRITY) : وهى القدرة على المحافظة على سلامة DB2 و سلامة البيانات وجعلها أو إبقائها فى حالة آمنة و مستقرة و تجنيبها الحالات الخطرة والضياع .

٣ – دليل DB2 :

يحتوى DB2 بشكل أساسي على نوعين من قواعد البيانات ، و هما قواعد بيانات المستخدم User databases ، وقاعدة بيانات الدليل catalog database، حيث تستخدم الأخيرة من قبل db2 لتوجيه تشغيل النظام والتحكم به، حيث يخزن داخل هذه القاعدة وصفا عن كل البيانات وبنائها التركيبي أو هياكلها التى يديرها قاعدة البيانات db2.

حيث يحتوى أو يتألف هذا الدليل من جداول تماما كما هو الحال مع قاعدة بيانات المستخدم، هذا بالإضافة أنه يمكن أن يطبق عليها استفسارات بعبارات SQL (و لكن لا يجوز تعريفها أو تعديلها) ، تماما مثل جداول المستخدم، حيث يمكن أن يستخدم الدليل هذا كقاموس بيانات وظيفي فعال من قبل أجزاء مختلفة من نظام DB2 و ذلك لأغراض أمنية أو لتحسين أداء النظام أو لاختزال استفسار SQL (حيث أن القاموس عبارة عن بنية أو هيكل للبيانات تحتفظ بمعلومات عن قاعدة البيانات الحالية).

mR

حيث يمكن أن يتألف دليل DB2 من ثلاثين جدول تقريبا منها:

- SYSTABLES : و هو جدول يحتوى على سطر من المعلومات من اجل كل جدول فى النظام .
- SYSCOLUMNS : و هو جدول يحتوى على سطر من المعلومات عن كل عمود من كل جدول من جداول النظام .
- SYSINDEXES : و هو جدول يحتوى على سطر من المعلومات عن كل فهرس مخصص لأي جدول فى النظام .

علما أن معظم برامج RDBMS ، إن لم نقل جميعها لها دليل مشابه أو قريبا منه .

- **برامج متعلقة بالنظام DB2 :**

برامج هذا النظام تتألف بشكل أساسي من عدة طبقات نواتها (kernel) .
حيث إن وظيفة هذه النواة تكمن فى إدارة البيانات، قبول أو رفض طلبات الوصول إلى البيانات وتمريرها بين قاعدة البيانات وواجهة المستخدم User Interface.
إن اللغة الوحيدة المفهومة من قبل نواة DB2 هي SQL و لهذا على أي مستخدم يحاول الوصول إلى البيانات المخزنة فى قاعدة بيانات DB2 أن يتصل معها بعبارات SQL.

- أما المنتجات البرمجية المتعلقة بالنظام القاعدةDB2 فهي كما يلي :

- DB2II (واجهة نظام DB2 التفاعلية

DB2 Interactive Interface) :

وهو عبارة عن واجهة بمثابة الصديق للمستخدم تتألف من قائمة رئيسية و سبعة خيارات تسمح للمبرمج و المدير المشرف بإنجاز العمل بشكل تفاعلي في بيئة db2 .

- QMF (برنامج إدارة الاستفسار query management facility) :
وهو عبارة عن برنامج بواجهة أمامية لكتابة التقارير ويعتبر وسيلة هامة لمستخدم النظام لإنجاز عمله، يتألف QMF من مجموعة قوائم و خيارات مقادة بالمفاتيح سهلة الأتباع و الاستخدام تؤمن وصول المستخدم إلى sql و db2 .

- DXT (مستخرج البيانات data extract) :
وهو عبارة عن برنامج لنسخ البيانات يسمح لنا بتحويل صيغة الملفات (التتابعية ، المفهرسة ، أو ذات البنية الهرمية) إلى صيغة جاهزة للتحميل في جداول db2 .

- DBEDIT :
وهو برنامج تحرير واجهات المستخدم.

- VSAM TRANSPARENCY :
وهو برنامج خدمي مصمم لجعل ملفات بيانات نظام DB2 قابلة للوصول إليها من قبل البرامج التطبيقية التي تتوقع وجود البيانات مخزنة على شكل ملفات VSAM (وهي ملفات تعتمد طريقة الوصول التتابعي الوهمي إليها virtual sequential access method) .

- DB2I :
وهو برنامج بواجهة أمامية مقاد بالقوائم وسهل الاستخدام، يعمل هذا البرنامج تحت إشراف TSO، حيث يزودنا DB2I بإمكانية المعالجة التفاعلية و بإمكانية تنفيذ سبعة برامج خدمية هامة من DB2 وهي :

MR

١. SPUFI (معالج SQL باستخدام دخل الملف SQL Processor Using File Input) وهـــو برنـــامج تنفيـــذ عبارة SQL التفاعلية .

٢. DCLGEN (مولد التصريحات Declarations Generator) :
وهو مولد وصف بيانات البرنامج التطبيقي.

٣. BIND : و هو مترجم SQL المضمن
(Embedded SQL Compiler)

٤. PROGRAM PREPARATION : و هو البرنـــامج التطبيقـــي لعبارات SQL المترجمة مسبقا .

٥. RUN : و هو منفذ واجهة البرنامج التطبيقي الأمامية .

٦. DB2 COMMANDS : و هو برنامج تفاعلي لإصدار أوامر DB2 (لا يجوز الخلط بينها و بين عبـــارات SQL) . يســـتخدم هـــذا البرنامج بشكل رئيسي من قبل مشغل النظام system operator و DBA (المـــدير المشـــرف لقاعـــدة البيانـــات (Database Administrator)) .

٧. DB2 UTILITIES : و هو برنامج خدمي لتشغيل خـــدمات DB2 للمحافظة على سلامة قاعدة البيانات .

إن معظم خيارات هذه البرامج موجهة لمبرمجي التطبيقات بعبارات SQL و للمدير المشرف لقاعدة البيانات DBA أو لمبرمجي النظام المسـئول عـــن صيانة قاعدة البيانات. وهى لا تهمنا الآن أو خارج حدود الكتاب.

ثانيا : نظام قاعدة البيانات ORACLE

حيث يعتبر نظام قواعد بيانات علائقية موزعة Distributed من إنتاج شركة ORACLE من Belmont California .

ويوجد من هذا النظام إصدارات تعمل على الحاسبات الكبيرة والشخصية، أطلقت الشركة الإصدار الأول منه عام 1977 و حاليا وصل إلى الإصدار 8.x منه، يتميز محرك SQL فيه بأنه مكتوب باللغة C، ونظرا لشعبيته هذه اللغة وقابلية نقلها فإن ORACLE متوفر على أكثر من ستين حاسب و نظام تشغيل مختلف. والمزايا الأساسية والمضافة لنظام ORACLE موجودة ومتوفرة لجميع هذه البنى المختلفة مما يوفر توحيدا و توفيرا لموارد شركات أنظمة معالجة البيانات المختلفة.

- **برامج متعلقة بالنظام ORACLE :**

تميزت منتجات ORACLE من لحظة وصولها إلى المستخدم بتنوعها و فعاليتها وهى

* SQL * PLUS : وهى نسخة ORACLE من اللغة SQL وهى تتفوق على نسخة IBM بالعديد من الوظائف والعوامل(OPERATORS).
* SQL * FROMS : وهى حزمة برمجية لتطوير التطبيق تسمح لنا ببناء التطبيق بإنشاء الشاشات أو الأشكال FORMS التي سيتعامل معها المستخدم، بالإضافة إلى بعض آليات التحكم و المحافظة على تجانس المعطيات ضمن الشاشات نفسها.
* SQL * CALC : وهى جريدة إلكترونية spreadsheet تربط مع قاعدة بيانات ORACLE وتعمل بشكل مشابه لجداول LOTUS 1-2-3 الإلكترونية .

MR

* SQL * GRAPH : وهى واجهة رسومية Graphics Interface لإظهار نتائج الاستفسار.

* SQL * MENU : وهى أداة تطوير لتطبيق ORACLE .

* EASY * SQL: وهو برنامج مرتكز على الأشكال FORMS يسمح لمستخدم النظام بالوصول إلى معلومات الجدول دون كتابة عبارات SQL.

* SQL * NET : وهو برنامج لتصالات شبكة ORACLE .

* SQL * STAR : وهو برنامج ORACLE لقاعدة البيانات الموزعة الذي يمكن من توزيع Distribute أنظمة ORACLE وغير ORACLE .

* QMX : وهو برنامج كتابة التقارير / الاستفسارات لمستخدم النظام وهو يحاكى برنامج qmf من IBM

* PRO * SERIES : وهى مترجمات SQL القابلة للتضمين Embedded SQL Precompilers من ORACLE والمتوفرة حاليا للغات PASCAL, COBOL, FORTRAN , ADA, BASIC, C, PL/1 .

ونتيح الـ ORACLE لمبرمج التطبيقات الجاد أدوات كثيرة مفيدة، كما تعتبر نسخة ORACLE من اللغة SQL النسخة الأقوى والأكثر فعالية في سوق البرامج، فهي تمنحنا إمكانية الربط الخارجي (Outer-Join) والبنية الهرمية (Hierarchical Structure) و تنسيق الإخراج وعوامل الوقت والتاريخ وكم كبير من توابع الأسطر (Row Function) .

ثالثا : نظام قاعدة البيانات INGRES

وهو نظام قواعد بيانات علائقية موزعة (Distributed) أيضا من إنتاج شركة RTI من Alameda California ، وأيضا يوجد من هذا النظام إصدارات تعمل على الحاسبات الكبيرة والشخصية، يتميز نظام INGERS بأنه مكتوب أصلا باللغة التجميعية للأنظمة التى يعمل عليها، وهو متوفر على أنظمة وحاسبات بشكل اقل من ORACLE .

- **برامج متعلقة بنظام قاعدة البيانات INGRES :**

تتميز أدوات INGRES بكونها تعمل من خلال شاشات شبيهة بقوائم LOTUS 1-2-3، مما يجعله سهلاً ومريحا جدا لمستخدمي النظام خلال عملهم، ويعود ذلك إلى أن استراتيجية تطوير INGRES تعتمد على قيادة معالجة البيانات والوصول إليها من خلال الأشكال والقوائم بشكل أكثر من DB2, ORACLE .

- **ما أدوات التطوير و صنع القرار فى INGRES فهي :**

- ISQL : و هو شبيه بنظيره SQL * PLUS من ORACLE باستثناء دعمه لإمكانية التحرير editing بالشاشات.

- INGRES / MENU : وهو الواجهة الأمامية التى تقود كافة وظائف وأدوات INGRES.

- INGRES /4GL : وهو مولد التطبيق باعتماد لغة الجيل الرابع Fourth Generation Language.

- VIFRED : و هو محرر الأشكال المرئي Visual Forms Editor لإنشاء الواجهة الأمامية للتطبيقات و طلبات الوصول إلى البيانات.

- VQL : لغة الاستفسار المرئيـة visual query language وهـى أداة بواجهة أمامية لنقل المعطيات.

- ABF (Application by Forms) :
وهى أداة توليد التطبيق وهى أيضا ذات واجهة أمامية.

- QBF (QUERY BY FORMS) :
وهى أداة ذات واجهة أمامية وهى تـربط مـع معطيـات INGRES والأدوات السابقة أيضا.

- RBF (REPORT BY FORMS) :
وهو كاتب تقارير INGRES و هو يعتمد واجهة أمامية أيضا.

يتميز INGRES ببعض المزايا المهمة كقيوده الفريدة على تجانس البيانـات وتنفيذه لحركة اليومية وهى مزايا غير متوفرة فى الأنظمة الأخرى فضـلا عـن سهولته البالغة بالنسبة للمستخدم بشكل عام.

أخيرا، يجب أن تعلم أن هناك بعض الاختلافات الثانوية فى صياغة عبارات SQL فى الأنظمة IGRES, ORACLE, DB2 ، وبالطبع تحتاج المراجعـة والتدقيق عند التنفيذ.

====================

الفصل الثانى

مدخل نظرى للغة SQL

Introduction to
Structured Query Language

يتناول هذا الفصل المحتويات الآتية :

- التعريف بلغة SQL
- إمكانيات لغة SQL
- الجداول
- الفهارس
- الملخصات
- قواعد البيانات

MR

تمهيد :

من خلال هذا الفصل سنتعرف على بعض أساسيات لغة SQL كلغة أساسية للتعامل مع قواعد البيانات وكأساس لها، حيث تعتبر المدخل الرئيسي لكثير من النظم المرتبطة بقواعد البيانات، بالإضافة إلى أن كتابنا هذا سيتناول بالشرح إحدى هذه النظم والمرتبطة بقواعد البيانات العلائقية Relational Databases وهو نظام أو قاعدة البيانات MySQL.

وتعتبر أساسيات البرمجة لهذه اللغة مدخل رئيسي لنظام My SQL ، ولذا فيمكن أن نتعرف على بعض عناصر هذه اللغة الأساسية، حتى يمكن الاستفادة منها فيما بعد وأيضاً عمل المقارنة فيما بينهما.

ولهذا فيجب على قارئ هذا الكتاب أن يمر بسرعة على هذا الفصل.

٢ – ١ التعريف (ما هية) بلغة SQL ؟

WHAT IS SQL LANGUAGE?

تعتبر لغة SQL اختصارا للعبارة

Structured Query Language

التى تعنى لغة الاستفسار الهيكلية، وهى التى تعنى ببساطة لغة للوصول (Access) إلى البيانات ومعالجتها وهى كسائر اللغات الأخرى تستطيع الاتصال مع غيرها.

- **مع من نتصل عندما نستخدم اللغة SQL ؟**

تتبادل SQL الاتصال مع مدير قواعـد البيانـات (Database Manager) وهو يمكن أن يكون DB2 أو SQLBASE أو ORACLE أو INFORMIX أو INGRES أو SQL Server أو MySQL أو أي نظام قواعد بيانات علائقي (RDBMS اختصارا) يدعم اللغة SQL.

هذا وتتعامل جميع هذه الأنظمة مع اللغة SQL و لذلك كلما أتقنت التعامل مع هذه اللغة كلما وجدت نفسك قادرا على تنفيذ عمليات أكثر تعقيدا فى معالجـة البيانات واستخلاص النتائج المرجوة.

بالإضافة لهذا فتتميز SQL بأنها مرنة وسهلة التعلم، لكثير من المسـتويات، فعلى الرغم من كونها لغة لبرمجة الحاسب الآلي إلا أنها أكثر بساطة من لغـات البرمجة الشهيرة مثل BASIC أو PASCAL أو C ،

- **ويمكن تعلمها بسهولة لعدة أسباب منها ما يلي :**

أولا : أن أوامر أو جمل أو عبارات SQL مكونة من المرادفات الإنجليزية لعمل أو وظيفة كل أمر (و هي خالية من الصيغ الرياضية أو الهندسية) ، على سبيل المثال:

- نختار بيانات بأمر أو بالكلمة (select) من الجداول بعبارة
SELECT ..... FROM

- عندما ننفذ شرط بأمر (where) تحقق شرط محدد بالفقرة WHERE،
- عندما ندرج أو نحشر قـيم معينـة فـى جـداول البيانـات بالكلمـات VALUES و INSERT،

MR

- أو نلغى البيانات من الجداول بالكلمات DELETE FROM عند تحقق شرط معين، وهكذا بالنسبة للأوامر أو الجمل الأخرى، لهذا ستجد اللغة منطقية جدا وسهلة الاستخدام ويمكن التعامل معها بسهولة.

ويمكن القول أن لغة SQL هى:

- لغة ذات فعالية عالية للوصول (Access) إلى البيانات، وتتميز بسهولتها ومرونتها فى التعامل.
- لغة علائقية قياسية Relational Standard Language.
- لغة استفسار (Query Language) قريبة جدا من اللغة الإنجليزية، و خالية من الصيغ الرياضية المختلفة .
- SQL مدعومة من قبل أنظمة مختلفة شهيرة منها ما يلي :

Oracle	db2	Sql / DS	Sql base
Informix	QMF	QINT	VAX Rdb
SQL Server	Access	MySQL	...

ثانيا : هذه اللغة تحدد أو تقرر بنفسها كيف ستجلب البيانات من وإلى قاعدة البيانات فكل ما عليك هو أن تحدد ما تريده من عمليات وستقوم SQL بالباقي، وتسمى هذه الطريقة أو الأسلوب بتقنية عدم الإبحار (no navigational)، وهى تقنية متطورة وعالية فى عالم تشغيل البيانات، هذا بالإضافة إلى أنه يمكنك أو يكفيك أن تعرف أن العديد من البرامج المكتوبة بأنظمة قواعد البيانات التجارية يمكن استبدالها بعبارة استفسار SQL وحيدة ومحددة.

ثالثا : أن معظم أنظمة DBMS تدعم المعالجة باللغة SQL بشكل إستاتيكي (static) أو تفاعلي (interactive)، وبالتالي يمكن الاستفادة من خبرات وعمل الآخرين.

لذا يمكن تنفيذ جمل أو عبارات SQL بشكل تفاعلي بالتحدث مباشرة إلى مدير قواعد البيانات أو من خلال تضمينها داخل لغات البرمجة التجارية كلغة visual basic أو DELPHI، وتعتبر هذه الميزة ضرورية جدا، فهذه اللغة صممت فى الأصل لتستخدم مع لغات البرمجة الأخرى، فاللغة SQL بذاتها لا تحتوى على أوامر لإظهار صناديق حوار(dialog) على الشاشة، ولا على أوامر تستطيع تنسيق أكثر من تقرير بسيط.

ولهذا فأن هذه الميزة الثنائية هامة جدا فى أي تطبيق معالجة أو تشغيل بيانات يتصف بشكلي (formal) .

فمثلا من خلال عبارات SQL المضمنة نستطيع الوصول إلى البيانات فى أنظمة ORACLE بطريقة تفاعلية أو ضمن البرامج بشكل دفعي – متتالي الأوامر (batch) ، حيث تعتبر عبارات SQL المضمنة نفسها شبيهة جدا بنظيراتها التفاعلية ، ويتضح ذلك من خلال دراسة العبارات المضمنة والتي تدعم اللغتين visual basic وDelphi للغة SQL .

رابعا : تشغل هذه اللغة البيانات بمستوى المجموعة (set level) وهذا يعنى أن التعديلات التى نجريها ستغير أو تؤثر على مجموعة من السجلات (records أو الأسطر (rows) داخل الجداول، وإن ناتج استفسار Query ما سيكون على شكل مجموعة من السجلات حسب الشروط المطلوبة.

ويتضح أن هذا يخالف نمط العمل المسمى سجل ... سجل ...record1...record2....record3 وذلك فى معالجة ملفات البيانات، وعلى الرغم من فعالية وسهولة الأنظمة العلائقية إلى أن مبدأ المعالجة " بمستوى المجموعة " يضع أمام أي شخص اعتاد المعالجة بالنمط record1...record2...record3 عدة عبارات.

MR

عندئذ يمكن أن نوضح أو نشرح الفرق بين هذين النمطين لنرى بوضوح فائدة نمط مستوى المجموعة من خلال المثال التالي ، كما تعلم فإن نمـــط العمـــل : record1...record2....record3 فى معالجة بيانات الملفات حيث يرتكز على الحلقة (loop)، ويتضح ذلك من خلال المثال التالي الذي يبين كيف يمكن زيادة حقل الســعر (price field) فـــى الســـجلات الخاصـــة بمنطقـــة مـــا ولـــيكن مدينة damietta :

```
Readloop;
  If EndOf file
       Goto Endloop
   Else
      Read Record
      if Record.SalseRegin = 'damietta' Then
      price = price*2.15
      write record
   Endif
 Goto ReadLoop
 Endloop
 End program
```

من خلال هذا نلاحظ أن هذا النمط قد تطلب منا إنشاء برنامج يقـــرأ ســـجل بيانات محدد أو وحيد، حتى يعالجه ثم يعود لقراءة سجل أخر جديد، وهكذا حتـــى ينجز يتم الانتهاء من العمليات أو الأعمال المرادة، وفى حالة أننا نعمل فى نمـــط مستوى المجموعة فالعملية الأخيرة أو عمليات التشـــغيل الســـابقة تنفـــذ علـــى مجموعة السجلات ككل، وليس على كل سجل واحد على حدة، فبدلا من استخدام حلقة تكرارية للبحث الدوري عن كل سجل، يمكن للغة SQL أن تنفذ العمل على كامل الجدول بعبارة واحدة، وذلك على النحو التالي :

```
UPDATE area SET price * 2.15
        WHERE area.salseRegin = 'damietta'
```

٢ – ٢ إمكانيات لغة SQL :

هذه اللغة لها قدرات على عمليات التشغيل SQL Processing capabilities للبيانات ، وتتكون هذه اللغة من لغة تعريف للبيانات، ولغة معالجة للبيانات، وأيضاً لغة تحكم بالبيانات، هذا بالإضافة إلى أنها تؤمن هذه اللغات الثلاث إمكانية دعم المعالجة العلائقية للبيانات، ومن هنا نجد أن معظم عمليات الوصول (Access) إلى البيانات فى الأنظمة المرتكزة على SQL هي من خلال SQL نفسها ومن خالها أو تعتمد عليها.

على سبيل المثال:

SQL Language	
	١ – فى لغة تعريف البيانات
CREATE	أمر عملية الإنشاء
ALTER	أمر تعديل الجداول
DROP	أمر إلغاء هياكل البيانات
	٢ – فى لغة معالجة البيانات
SELECT	أمر عرض البيانات
UPDATE	أمر تعديل البيانات
DELETE	أمر حذف السجلات
INSERT	أمر إضافة البيانات
	٣ – فى لغة التحكم بالبيانات
GRANTE	إعطاء صلاحيات للمستخدمين
REVOKE	إلغاء صلاحيات

MR

وهنا يقصد بتعريف البيانات (Data Definition) إمكانية إنشـاء (Create) بناء هياكل البيانـات (Data Structures) أو إلغـاء (Delete) أو تعـديل (Modify) البيانات المطلوبة للنظام ، حيث يتضمن هذه البناء قواعـد البيانـات (Databases) والجداول (Tables) و الفهارس (Indexes) وهى الأقسام التى تمثل هياكل البيانات فى نظام قواعد البيانات .

وأيضاً يمكن أن تقسم عملية المعالجة إلى ثلاثة أقسام رئيسية هي:
استرجاع البيانات (Retrieving Data)
معالجة البيانات (Manipulating)
وتعديل البيانات (Updating)

حيث نقصد بالاسترجاع الاستفسار (Queering) فى قاعدة البيانـات وهـو اختيار بيانات مخزنة فى الجداول لتقديمها على الشاشة.

ويقصد معالجة البيانات فهي تشير إلى إمكانية إنجاز توابع أو دوال إحصائية على البيانات كالمتوسط الحسابي أو مجموع الأعمدة أو إجراء عمليـة الضـرب عليها.

وأما تعديل البيانات فهو يشير إلى إمكانيـة إدراج أو إلغـاء الأسـطر فـى الجداول وتغيير القيم فى الأعمدة أي إصلاح البيانات.

ومن ناحية التحكم فى البيانات فهو يسمح لنا بتعريف آلية أو طريقـة سـرية (Security Mechanism) لحماية البيانات فى الأنظمة من الوصول إليها مـن قبل غيـر المرغـوب فـيهم، ويتضـمن ذلـك إمكانيـة تخويـل - إعطـاء صـلاحيات (Granting) و أبطـال (Revoking) صـلاحيات معالجـة (Processing Privileges) بيانات النظام لمستخدمين آخرين لهـذا النظـام،

ولهذا نجد أن عبارات الوصول إلى البيانات تسمح بوضع إمكانية تحكم بسيطة أو معقدة لتحقيق ذلك بسهولة.

وتمتاز أيضاً عبارات SQL ميزة أخرى هامة وهى إمكانية تنفيذها ديناميكيا (DYNAMIC)، وهذا يعنى إمكانية تنفيذها فى أي وقت يعمل فيه النظام، على سبيل المثال إذا أردت إلغاء قاعدة بيانات موجودة وتعريف واحدة أخرى جديدة ستجد أن تحقيق ذلك لن يتطلب مجهود كبير، ولذا سنجد تعريف قواعد البيانات العلائقية بسيط وسهل التعديل عند تعريفه وهكذا.

٢ - ٣ الجداول (TABLES) :

توجد أو تنظم البيانات فى أنظمة قواعد البيانات العلائقية وتقدم على شكل جداول (انظر الشكل التالى) تشبه هذه الجداول تلك التى نتعامل معها فى حياتنا اليومية (كجداول مواعيد السفر أو كدليل الهاتف و قوائم أسماء معينة والتواريخ لمواعيد معينة الخ ...)، ولهذا يمكننا اعتبار الجدول تجميعا لأسطر (ROWS) وأعمدة (COLUMNS) متعلقة ببعضها تصف كينونة (Entity) عمل فريد كقسم (Department) أو مشروع (Project) أو معلومات عن قطع يجرى تزويدها أو حوالات شراء (Purchase Orders) أو طلبات تأمين (Insurance Order)..الخ.

و كمثال يمكننا النظر إلى بيانات الطلاب كجدول يحتوى سطرا وحيدا لكل طالب أو طالبة ، بحيث تخزن المعلومات المختلفة فى كل سطر على شكل أعمدة يصف كل عمود منها رقم الطالب، اسم الطالب، تاريخ الميلاد النوع المكافأة.

المكافأة	النوع	تاريخ الميلاد	الاسم	رقم الطالب
٥٠٠	ذكر	١٩٨٠/١/١	محمد	١
١٠٠	انثى	١٩٩٠/٦/٥	منى	٢
٣٠٠	ذكر	١٩٩١/٥/٤	على	٣
٤٠٠	ذكر	١٩٨٩/١/١	احمد	٤

صف (سجل) Row (Record)

حقل (خلية) Field

عمود Column

جدول Table

*** وتتكون الجداول من :**

- الأسطر Rows:

وهى العناصر الأفقية من الجداول (انظر الشكل السابق) ويمكنك ملاحظة أنها غير مسماه وهى غالبا غير مرتبة، هذا خلافا للأعمدة، وهذا يعنى انــه لا توجد طريقة لتحديد سطر معين من الجدول من خلال معرفــة موقعــه ضــمن الجدول(أي السطر الأول، أو الأخير، أو ....) إذ أن الوصول (Access) إلى الأسطر يتم عن طريق قيم البيانات فقط التى به.

أي يمكن لأي شخص يتعامل مع قواعد البيانات أن يلاحظ أن الأسطر تكافئ تقريبا مبدأ السجل فى أنظمة التشغيل التقليدية، أي مجموعة واحدة من القيم تقرأ أو تكتب كوحدة متكاملة، وعلى هذا يكافئ الجدول ملفا تقليديا فى نظم التشــغيل التقليدية مثل DBASE أو ACCESS , FOX.

١ - وهنا يمكن القول أن الأسطر تتألف من مجموعة من القيم (قيمة واحدة لكل عمود)

٢ - وأيضاً تتميز بأنها:
أ - غير مسماه.
ب - غير مرتبة.
ج - يتم الوصول إليها عن طريق قيم البيانات التى تحويها.

٣ - السطر يعادل السجل فى أنظمة قواعد البيانات التقليدية.

٤ - أما الجدول فيعادل ملف فى أنظمة قواعد البيانات التقليدية.

- الأعمدة Columns :

وهى عناصر البيانات فى الجدول والتي تكون بالشكل الرأسي فى الجدول ، وخلافا للأسطر نجد الأعمدة مسماة كما يمكن ترتيب عرضها بعبارات SQL كالتي تعرف الجدول (العبارة Create) و هذا يعنى بالطبع إمكانية تحديد موقعها (من اليسار إلى اليمين) ضمن عبارات الاستفسار.

- حيث يكون لكل عمود فى الجدول اسما فريدا (Unique Name) ضمن الجدول.

ويحدد له أيضا صفة مميزة (Attribute) لقيم البيانات التى ستسجل فيه. تفيد الصفة المميزة للبيانات فى وصف ثلاثة أشياء هي:

١. الطول أو الحجم للبيانات التى ستقع فى العمود.
٢. نوع البيانات المخزنة فى العمود (هل هي أبجدية –نصية– أم عددية فقط).

MR

٣. أيضا تصف إمكانية أن تكون قيم البيانات غير معروفة (Null Values) تعامل SQL هذه القيم معاملة خاصة مميزة عن غيرها وهى خاصة بهذه اللغة.

ويمكن القول أن:

- الأعمدة هي مجموعة حقول مسماه.
- الأعمدة هي عناصر الجدول الرأسية.

= يتم إنشاء الأعمدة عن طريق :

- تعريف اسم العمود .
- تحديد الصفات المميزة للبيانات التى يحملها.
- تحديد الحجم الخاص به.

- الحقول Fields :

يحتوى الحقل إلى قيمة محددة من البيانات المخزنة ضمن الجدول تقع عند تقاطع عمود و سطر، تتعامل SQL مع الحقل على انه أصغر عنصر بيانات فى الجدول، لأنه يمكن أن يحتوى على رقم أو حرف أو أكثر من ذلك.

- الحقل يشير إلى قيمة للبيانات تقع عند تقاطع سطر وعمود محددين.
- الحقل يمثل اصغر وحدة بيانات تستطيع SQL معالجتها والتعامل معها.
- الحقل عندما يحتوى على null يشير إلى عدم وجود قيمة للبيانات فيه.



٢ - ٤ الفهارس INDEXES :

الفهارس تعتبر مؤشرات (Pointers) إلى أسطر معينة (سـجلات بيانـات) ضمن الجدول، وهى مرتبة منطقيا عن طريق قيم مفتاح ما فمثلا سيخزن الفهرس المنشأ بالمفتاح LAST-NAME جميع أواخر الأسماء فى الجدول مع مؤشـر إلى أسطرها المتعلقة بها، وهى عملية متماثلة لعملية استخدام فهرس كتاب مـا، فإذا أردت مثلا إيجاد الصفحات التى أشارت إلى العبـارة Order By كأحـدي عبارات SQL ضمن الكتاب نرجع إلى فهرس الكتـاب لنجـد سـردا بجميـع الصفحات المطلوبة عند الكلمة ORDER BY، وهكذا فنحن نذهب مباشرة إلـى الصفحات التى احتوت معلومات عن ORDER BY، بالطبع سيجعل ذلك البحث عن المعلومات عملية سريعة، لسوء الحظ فأن الفهارس تبطئ عمليـة التعـديل (Update) وهذا يفرض علينا التردد بعض الشيء قبل إجراء عملية الفهرسة.

وعلى الرغم من ذلك فإن فهرسة عمود مفتاح واحد أو اثنين من كل جـدول غالبا ما تكون جيدة، فهي تبسط وتسرع استرجاع البيانات بشكل جيد.
حيث يختلف نظام DB2 عن بعض أنظمة DBMS المستخدمة على أجهزة PC مثل DBASE ، ففى نظام DB2 يقرر النظام نفسه وليس المستخدم فيما إذا كانت هناك حاجة لاستخدام الفهرس فى تحديد موقع سطر معين داخل الجدول.
وأيضاً للفهارس ميزة أخرى هامة وهى أنها من الممكن استخدامها لفرض قيم مميزة على العمود أو بمعنى أخر إذا تم تعريف عمود (مثل عمود رقم البطاقـة الشخصية) له الخاصة أو الخيار Unique، عندها لن يسمح النظام بوجود قيمة مكررة تحت نفس العمود المحدد.

أى أن :

- تمثل الفهارس مؤشرات إلى سجلات البيانات فى الجدول وهـى مرتبـة منطقيا عن طريق مفتاح معين.

MR

- تنشأ الفهارس من قبل المستخدم.
- يحافظ DBMS بشكل دائم على الفهارس.

مميزات وعيوب الفهارس :

- تسرع استرجاع المعلومات
- قد تبطئ عملية التعديل (UPDATE)
- يمكن استخدامها لفرض قيم متمايزة لعمود ما
- تخزن الفهارس بشكل منفصل عن الجداول
- يمكن إنشاؤها بأحد الخيارين :

Unique
/ nonunique clustered

- يمكن إنشاؤها في أي وقت
- لا يشار إليها أبدا في عبارات SQL .

٢ – ٥ الملخصات VIEWS :

وهو تعريف مرتبط ببناء بيانات معينة، حيث يمثل مجموعة فرعية من الأسطر والأعمدة من جدول واحد أو أكثر، وبالنسبة للمستخدم سيبدو الملخص كأنه جدول، إلا أن الملخص صورة أو وصف لمعلومات الجدول ولا يخزن أية بيانات فيزيائية ولذلك تسمى الملخصات أحيانا " بجداول وهمية" (virtual tables) أو تخيلية(imaginary)

تتمتع الملخصات بإمكانية تعديل محدودة.
وتستخدم بشكل أساسي للأحوال أو الأمور التالية :

١. تبسط الملخصات عملية معالجة استفسارات المستخدم كونها تمثل نتائج استفسارات مكتوبة مسبقا .

٢. يمكن استخدامها كجزء من إجراءات أمن وسرية البيانات كونها تقيد الوصول (access) إلى أسطر / أعمدة معينة.

وهكذا يمكننا فهم المصطلح " ملخص" من وظيفته : أنــه يــزود المســتخدم (حسب دوره) بنظرة محددة ومقيدة على المعلومات المخزنة فى قاعدة البيانات . فمثلا قد يطلب مدير قاعدة البيانات ملخصا مؤلفا من العمود الأول والرابع مــن الجدول، ليشكل منها نظرة المستخدم A للجدول تبعا لتوجهات وعمل المستخدم A على الجدول، حيث قد لا يرى بيانات أخرى ضمن الجدول.

*** ويمكن القول أن :**

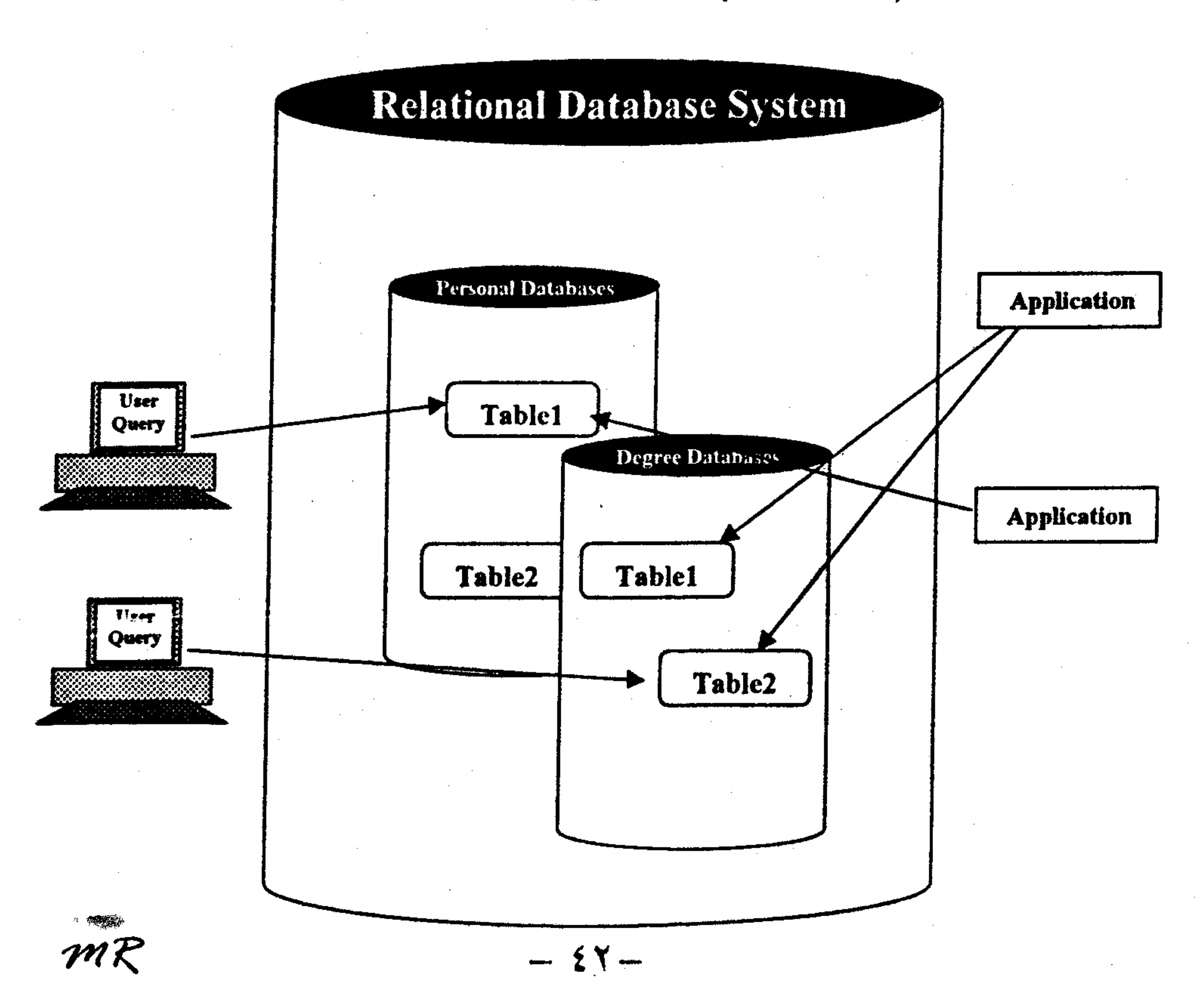
- يمثل الملخص جدولا وهميا.
- الملخص لا يمثل التخزين الفيزيائي – المادي للبيانات.
- الملخص يمثل مجموعة فرعية من الأسطر والأعمدة من جدول واحد أو أكثر.
- يسهل ويبسط الملخص معالجة الاستفسارات.
- يتميز الملخص بإمكانيات تعديل محدودة.
- الملخص يمنحنا إمكانية ترشيح البيانات قبل تقديمها للمستخدم.
- يمكن استخدام الملخص كجزء من إجــراءات أمــن البيانــات تخــزن الملخصات ضمن دليل نظــام (SYSTEM CATALOG) قاعــدة البيانات.

٢ –٦ قواعد البيانات DATABASES :

بصفة عامة قد يكون الفرق بين قاعدة البيانات ونظام إدارة قاعــدة البيانــات ولغة الوصول إلى البيانات محيراً بعض الشيء ولهذا سنوضح تعريفا لكــل واحد منهم.

- حيث تمثل قاعدة البيانات مجموعة متكاملة من الملفات أو الجـــداول المتعلقة ببعضها البعض.
- يمثل نظام إدارة قواعد البيانات (DBMS) كامل النظـــام المســئول عن إدارة ملفات قواعد البيانات المختلفة، بما فيه قواعد البيانات.
- حيث تمثل لغة الوصول إلى المعطيات SQL اللغة المستخدمة مـــن قبل المستخدم للاتصال مع DBMS لإنجاز الأعمال المطلوبة منه.

تكون أو تعمل قواعد البيانات فى الأنظمة العلائقية بشكل أساســي كمخـــازن (Repositories) لمجموعة من الجداول المتعلقة ببعضها. فمثلا يمكـــن لجميـــع جداول الموظفين (Personnel) أن تخزن كقاعدة بيانات واحدة للموظفين.



ويمكن أيضا على سبيل المثال :

أن تخزن جميع الجداول المحتوية على بيانات أو معلومـــات عـــن الرواتـــب (PAYROLL) فى قاعدة بيانات مستقلة خاصة بجداول المرتبات، وهكذا يضمن لنا التخزين والتجميع لكل ظاهرة على حدة.

وبالطبع تتم عمليات التشغيل أو المعالجة للبيانات من قبل البرنامج التطبيقـــي بشكل مناسب وسهل، لذا فالحفاظ على جداول قواعد البيانات العلائقية وفهارسها ضمن قاعدة بيانات واحدة، سيسمح لنا باستخدام أمر واحـــد لتحقيـــق المطلـــوب عوضا عن تنفيذ الأمر على كل جدول بشكل مستقل.

====================

الفصل الثالث

مدخل نظرى للتعرف على MySQL
Introduction to MySQL

يتناول هذا الفصل المحتويات الآتية:

- التعرف على نظام قاعدة البيانات MySQL
- مميزات نظام قاعدة البيانات MySQL
- إمكانيات وحدود نظام قاعدة البيانات MySQL
- العناصر المختلفة المكونة لجمل MySQL

٣ - ١ التعرف على نظام MySQL :

هو نظام جاهز أو برنامج تطبيقي جاهز خاص بنظم إدارة قواعد البيانات العلائقية RDBMS (Relational Data bases Management System) كأحدى أنواع نظم قواعد البيانات والمنتشرة الاستخدام في مجال التصميم لقواعد بيانات الويب Web، والتي نكون إحدى عناصر إنتاج الويب والتي يمكن أن تتعامل مع عدد من المستخدمين في نفس الوقت، وبالطبع الاستفادة من تزويدهم السريع ووصولهم إلى البيانات والمعلومات الخاصة بالموقع بسهولة وسرعة.

بالطبع هذا النظام يستعمل لغة SQL وإلى حد ما نفس الأوامر وهي اختصار (Structured Query Language)، لغة الاستعلام الهيكلية (البنائية) وهذا النظام أو القاعدة تمكن المستخدمين لها من تصميم قاعدة بيانات يمكن إجراء كثير من العمليات عليها، على سبيل المثال تصميم هياكل القاعدة، إضافة أو تخزين البيانات ، إجراء عمليات التعديل المختلفة على بيانات القاعدة ، إجراء عمليات البحث والفرز للبيانات ، إنتاج أو استخراج البيانات أو المعلومات بفعالية ومرونة.

ومن هنا يمكن القول أن في حالة معرفة إمكانيات هذا البرنامج واستخدامه، يمكن للقارئ ربطة بإحدى لغات برمجة تصميم صفحات الويب Web مثل لغة PHP أو ASP أو غيرها ، وبالتالي يمكن بسهولة تصميم مواقع ويب ديناميكية (على شبكة الانترنت بالطبع) ذات قدرات وإمكانيات وأمان ومرونة عالية تتصف بالمواقع التجارية الكبيرة ذات الحركة الديناميكية التى يمكن أن يستخدمها أفراد أو مستخدمين ذات قدرات مختلفة، وذلك باستخدام أنظمة أو برامج قواعد البيانات، خادمة مخصصة للربط على الشبكات مثل MySQL Server، وأيضاً هناك قواعد بيانات يمكن أن ترطبت بلغات برمجة المواقع (PHP أو ASP أو غيرها) مثل Informix, Sybase, MsSQL Server, PostgrSQL, Oracle, Hyperwave, dbm, filePro .

MR

٣-٢ مميزات نظام MySQL :

بالطبع لوجود أنظمة أخرى ذات نفس الطابع (قواعد بيانات) في السوق العالمي، سيكون هناك منافسة فيما بينها ، وبالتالي تبحث كل شركة أو منتج للنظام أو يوجد كثير من نقاط القوة داخل برنامجه، حتى يوجد مكان لبرنامجه داخل السوق العالمي، ولهذا فإن هذا النظام قد يمتاز ببعض نقاط القوة التي يمكن عرضها كما يلي:

١ – التكلفة منخفضة

حيث أنه فى الغالب أن كل منتج له تكلفة ما وهذا النظام يوجد شفرة البرنامج الأصلي له أو المترجم الخاص به فى شكل (المصدر المفتوح Open Source)، وبالتالي يمكن إيجاده بسهولة والترخيص له متاح بتكلفة قليلة بالنسبة للمشاريع التجارية .

٢ – السرعة العالية

يمتاز هذا النظام بسرعته العالية في التعامل معه ومع عناصر قاعدته بالمقارنة مع البرامج الأخرى.

٣ – مرونة الأداء

يتصف ويمتاز هذا النظام بالمرونة العالية فى التعامل معه، مع إمكانيات استخدام تتصف بالمرونة من قبل المستخدمين.

٤ – سهولة الاستخدام

يتصف ويمتاز هذا النظام بسهولة الاستخدام سواء أثناء التصميم للقاعدة من قبل المصممين أو من قبل المستخدمين لمنتجة بعد التصميم.

٥ – البرنامج (المترجم) الأصلي مفتوح المصدر Open Source

هذا النظام يوجد برنامجه المصدر بصوره مفتوحة يمكن التعامل معه والتعديل فيه فى أي وقت تحتاجه.

٦ – التعامل والتنقل مع أنظمة تشغيل مختلفة

هذا النظام يمكن أن يتعامل مع أنظمة تشغيل حاسبات مختلفة، بنفس القدرة، مثل نظم LINUX , WINDOWS.

٧ – تعدد المستخدمين

يمكن استخدام منتج هذه القاعدة من قبل أكثر من مستخدم في نفس الوقت، وبالتالي يتصف بالعمومية.

٨ – تعدد جداول القاعدة وسهولة الربط فيما بينها

يمكن إنشاء أكثر من جدول (ملف البيان) للقاعدة الواحدة ، وبالتالي إتاحة تسجيل بيانات مختلفة لكل منها مع إمكانية التعامل معها بسهولة.

٣ - ٣ إمكانيات و حدود برنامج MySQL version: 3.23.33 :

١- أنواع حقول البيانات الرقمية :

الوصف	التخزين (بايت)	النطاق	النوع
أرقام صحيحة صغيرة جداً	1	128 .. −127 أو 255 .. 0	TINYINT[(M)]
أرقام صحيحة صغيرة	2	32767 .. − 32768 أو 65535 .. 0	SMALLINT[(M)]
أرقام صحيحة متوسطة الحجم	3	8388607.. − 8388608 أو 16777215 .. 0	MEDIUMINT[(M)]
أرقام صحيحة اعتيادية	4	$1-2^{31}$ .. -2^{31} أو $1-2^{32}$ .. 0	INT[(M)]
مرادف لـ INT	4	مرادف لـ INT[(M)]	INTEGER[(M)]
أرقام صحيحة كبيرة	8	$1-2^{63}$ .. -2^{63} أو $1-2^{64}$ .. 0	BIGINT[(M)]

٢- أنواع حقول البيانات ذات النقطة العائمة:

	التخزين (بايت)	النطاق	النوع
	يختلف	يعتمد على precision	**FLOAT(precision)**
الوصف	يمكن استعماله لتحديد أرقام ذات نقطة عائمة فردية أو مزدوجة الدقة		
	4	±1.175494351E-38 ±3.402823466E+38	**FLOAT[(M,D)]**
الوصف	أرقام ذات نقطة عائمة فردية النقطة. هذه مرادفة لـ FLOAT(4) لكن مع طول محدد وعدة منازل عشرية		
	8	±1.7976931348623157E +308 ±2.2250738585072014E -308	**DOUBLE[(M,D)]**
الوصف	أرقام ذات نقطة عائمة مزدوجة الدقة. هذه مرادفة لـ FLOAT(8) لكن مع طول محدد وعدة منازل عشرية		

MR

DOUBLE PRECISION[(M,D)]	كالسابق DOUBLE[(M,D)]	كالسابق 8	
مرادف لـــ DOUBLE[(M, D)] .			**الوصف**
REAL[(M,D)]	كالسابق DOUBLE[(M,D)]	كالسابق 8	
مرادف لـــ DOUBLE[(M, D)] .			**الوصف**
DECIMAL[(M[,D])]	يختلف	M+2	
رقم ذو نقطة عائمة مخزن كـــ char . يعتمد النطاق على M وهو الطول المعروض.			**الوصف**
NUMERIC[(M,D)]	كالسابق DECIMAL[(M[,D])]	كالسابق M+2	
مرادف لـــ DECIMAL.			**الوصف**

٣- أنواع حقول بيانات التاريخ والوقت :

الوصف	النطاق	النوع
تاريخ. سيظهر كـ YYYY-MM-DD	1000-01-01 9999-12-31	DATE
وقت. سيظهر كـ HH:MM:SS لاحظ أن النطاق أكبر بكثير مما ستحتاج إليه أبداً.	-838:59:59 838:59:59	TIME
تاريخ ووقت. سيظهر كـ *YYYY-MM-DDHH:MM:SS*	1000-01-01 00:00:00 9999-12-31 23:59:59	DATETIME
ختم زمني. مفيد للمعاملات. يعتمد تنسيق العرض على قيمة M (راجع الجدول التالى رقم٤)	1970-0-01 00:00:00 2037	TIMESTAMP[(M)]
سنة. يمكنك تحديد تنسيق بـ ٢ أو ٤ أعداد. كل تنسيق من هذين التنسيقين له نطاق مختلف، كما هو مبين هنا.	69-70 (2069-1970) 2155-1901	YEAR[(2|4)]

MR

- أنواع TIMESTAMP والقيم التى يأخذها المتغير M :

النوع	الغرض
TIMESTAMP	YYYYMMDDHHMMSS
TIMESTAMP(14)	YYYYMMDDHHMMSS
TIMESTAMP(12)	YYMMDDHHMMSS
TIMESTAMP(10)	YYMMDDHHMM
TIMESTAMP(8)	YYYYMMDD
TIMESTAMP(6)	YYMMDD
TIMESTAMP(4)	YYMM
TIMESTAMP(2)	YY

٤- تنقسم أنواع السلاسل لثلاث مجموعات :

أولا : CHAR (حرف بطول ثابت) و **VARCHAR** (حرف بطول متغيـر)..
والفرق بين النوعين يكمن فى (مساحة التخزين والسرعة) .. بمعنى أن النوع CHAR عند تحديد مساحة ١٥ حرف له مثـل ((CHAR(15)) ثـم قمنا بتخزين كلمة مكونة من ٧ أحرف فقط، فإن باقى المسـاحة وهـى ٨

أحرف سيتم ملئها بفراغات !! أى أن هذه المساحة (٨ أحرف) ستكون بلا فائدة داخل الجدول ... أما VARCHAR فالوضع يختلف فعند تحديد (VARCHAR(15)) ثم قمنا بتخزين كلمة مكونة من ٧ أحرف فقط، فإن باقى المساحة وهى ٨ أحرف سيتم حذفها وسيسجل فى هذا الحقل فقط ٧ أحرف + واحد (للنظام) ... أى أن VARCHAR تحافظ على مساحة التخزين وتزيل الفراغات التى ليس لها فائدة أما CHAR فهى تضيع مساحة التخزين الزائدة دون فائدة ... هذا من حيث **المساحة التخزين** ... أما من حيث **السرعة** فـ CHAR أسرع من VARCHAR ... فى العمليات المختلفة.

- أنواع السلاسل الاعتيادية (حقول حرفية) :

النطاق	النوع
حرف 1 to 255	**[NATIONAL] CHAR(M) [BINARY]**
الوصف	سلسلة ثابتة الطول طولها M ، حيث M هو رقم بين ١ و ٢٥٥ . وتحدد الكلمة الأساسية NATIONAL أنه يجب استعمال مجموعة الأحرف الافتراضية، هذا هو الاختيار الافتراضى فى MySQL فى جميع الأحوال ، لكنه مشمول كونه جزء من المعيار ANSI SQL . وتحدد الكلمة الأساسية BINARY أنه يجب معاملة البيانات كغير حساسة لحالة الأحرف (علما بأن الخيار الافتراضى هو حساسة لحالة الأحرف).

[NATIONAL] VARCHAR(M) [BINARY]	1 to 255 حرف
كالسابق غير أنها متغيرة الطول .. أى أنك فى حالة كتابة VARCHAR(10) وقمت بتسجيل ٣ أحرف فى هذا الحقل فقط فإن البيانات التى سيتم تخزينها تساوى ٣ + ١ أى ٤ بايت فقط .. وليس ١٠ أى أن الحجم متغير مع البيانات التى سيتم تسجيلها.	**الوصف**

ثانياً : TEXT و BOLB .. تأتى هذه فى أحجام مختلفة، فهى تكون للنصـــوص الأطول أو البيانات الثنائية على التوالى.

والكلمة BLOB هى اختصار Binary Large Objects (الكائنات الثنائيـــة الكبيرة) فهى تستطيع تخزين أى شيىء تريد تخزينـــه، مثـــل الصـــور أو بيانات صوتية ...

وعملياً الأعمدة ذات النوع TEXT و BOLB هى نفسها ما عدا أن النوع TEXT حساس لحالة الأحرف أما BLOB فهو غير حساس. وهذه الأنواع تستطيع تخزين كميات كبيرة من البيانات.



- الأنواع TEXT و BLOB :

النوع	الطول الأقصى (أحرف)	الوصف
TINYBLOB	$1 - 2^8$ (أى 255)	حقـــل كـــائن BLOB صغير جداً
TINYTEXT	$1 - 2^8$ (أى 255)	حقل TEXT صغير جداً
BLOB	$1-2^{16}$ (أى 65,535)	حقـــل BLOB حجمـــه عادى
TEXT	$1-2^{16}$ (أى 65,535)	حقـــل TEXT حجمـــه عادى
MEDIUMBLOB	$1-2^{24}$ (أى 16,777,215) = 1.6 MB	حقل BLOB متوســـط الحجم
MEDIUMTEXT	$1-2^{24}$ (أى 16,777,215) = 1.6 MB	حقل TEXT متوســـط الحجم
LONGBLOB	$1-2^{32}$ (أى 4,294,967,295) = 4.2 GB	حقل BLOB طويل
LONGTEXT	$1-2^{32}$ (أى 4,294,967,295) = 4.2 GB	حقل TEXT طويل

ثالثاً: SET و ENUM .. تستعمل **SET** لتحديد أن القيم فى هذا العمود يجب أن تأتى من مجموعة معينة من القيم المحددة يمكن أن تحتوى قم الأعمدة على أكثر من قيمة واحدة من المجموعة ، ويمكـــن وضـــع ٦٤ عنصـــر فـــى المجموعة الواحدة ليتم الاختيار منهم (مثل مربعات الاختيار (القائمة)) ..

أما **ENUM** فهى اختصار Enumerated وتعنى تعداد ، وهى مشابهة جدا لـــSET، والاختلاف أن الأعمدة من هذا النوع يجب أن تحتوى (أى يتم اختيار) قيمة واحدة فقط أو NULL من قيم هذا العمود (مثل زر خيار).

MR

- الأنواع SET و ENUM :

الوصف	القيمة القصوى في المجموعة	النوع
تستطيع الأعمدة (الحقول) من هذا النوع تخزين فقط إحدى القيم المذكورة أو NULL	65535	ENUM('value1','value2', ...)
تستطيع الأعمدة (الحقول) من هذا النوع تخزين مجموعة من القيم المحددة أو NULL	64	SET('value1','value2', ...)

٣-٤ العناصر المختلفة المكونة لجمل MySQL :

* يحتوى جمل هذا النظام على عناصر (أوامر) خاصة بالبرمجة تصنف كما يلى:

١ - الأوامر COMMANDS

على سبيل المثال :

Create database	إنشاء قاعدة البيانات
Create Table	إنشاء جدول داخل القاعدة
Drop	حذف قواعد البيانات أو الجداول
Delete from	حذف سجلات من الجداول
Select ... from	اختيار أو انتقاء سجلات من الجداول
Update	التعديل للقيم داخل سجلات الجداول
Alter Table	التعديل فى هيكل الجداول
Insert into	الفهرسة داخل الجداول

٢ - الدوال FUNCTIONS

على سبيل المثال :

SUM	إيجاد المجموع بيانات داخل الجدول
AVG	إيجاد المتوسط
MAX	القيمة الأقصى
MIN	القيمة الأدنى
COUNT	الحصر للعدد

٣ – المقاطع CLAUSES

وهى مرتبطة بتكوين مقاطع معينة داخل الجملة، وبالطبع تربط بـأكثر من أمر من أوامر النظام او تحتوى على أوامر مرتبطة بتنفيذ عمليات معينـة، على سبيل المثال تكون مع :

ORDERD BY	أمر ترتيب البيانات حسب حقل معين
GROUP BY	

٤ – المعاملات OPERATION

أ – معاملات الحسابية Arithmetic Operation

على سبيل المثال :

+	الجمع
–	الطرح
*	الضرب
/	القسمة

ب – معاملات المنطقية Logic Operation

مثل :

لتحقيق شرطين	AND
لتحقيق أحد الشرطين	OR
عدم تحقيق الشرط	NOT

ج – المعاملات مقارنة Comparison Operation

مثل :

أكبر من	>	أصغر من	<
أكبر من أو يساوى	>=	أصغر من أو يساوى	<=
يساوى	=	لا يساوى	<>

– بالإضافة إلى معاملات أخرى

مثل :

بين	BETWEEN
تشبه أو تماثل	LIKE
فى	IN

وبالطبع تعتبر هذه المكونات حدود لعمل ومكونات البرنامج وفى نفس الوقت عناصر قوة للبرنامج، وسيتم التعرف على بعض هذه الوحدات فيما يلي، وهــى تعتبر ملفات مستقلة لها طبيعة مستقلة، ويمكن إجراء عملية الــربط فيمــا بينهــا والاستفادة من مميزات كل منها .

====================

الفصل الرابع

الإعداد والتشغيل
لبرنامج MySQL
version: 3.23.33

يتناول هذا الفصل المحتويات الآتية :

- **الإعداد والتهيئة**
- **التشغيل**
- **الخروج من MySQL والإغلاق النهائى له**
- **إزالة *MySQL* بالكلية من الـ *Windows***

MR

٤-١ الإعداد والتهيئة :

لإعداد قاعدة البيانات MySQL نحتاج بالطبع لتوافر جهاز حاسب آلـــى ذو إمكانيات معقولة ... أى انه يفضل آلا يقل نوع الجهاز عن Pentium II وذلك للتعامل بكفاءة وسرعة مع أوامر قاعدة البيانات هذه.

* فمثلا يمكن اتخاذ جهاز حاسب آلى بالمواصفات الآتية:

١ - ذاكرة رئيسية RAM لا تقل عن 16 MB ويفضل الأعلى .

٢ - شاشة عرض يفضل VGA أو SVGA بالإضافة إلى كارت الشاشـــة من نفس النوعية .

٣ - وحدة إدارة ثابتة HARD DISK يفضل ذو سعة عاليـــة ، (يفضـــل الأنواع ذو السعة التى تقاس بالـــ GEGA) .

٤ - وحدة إدارة أقراص الليزر.

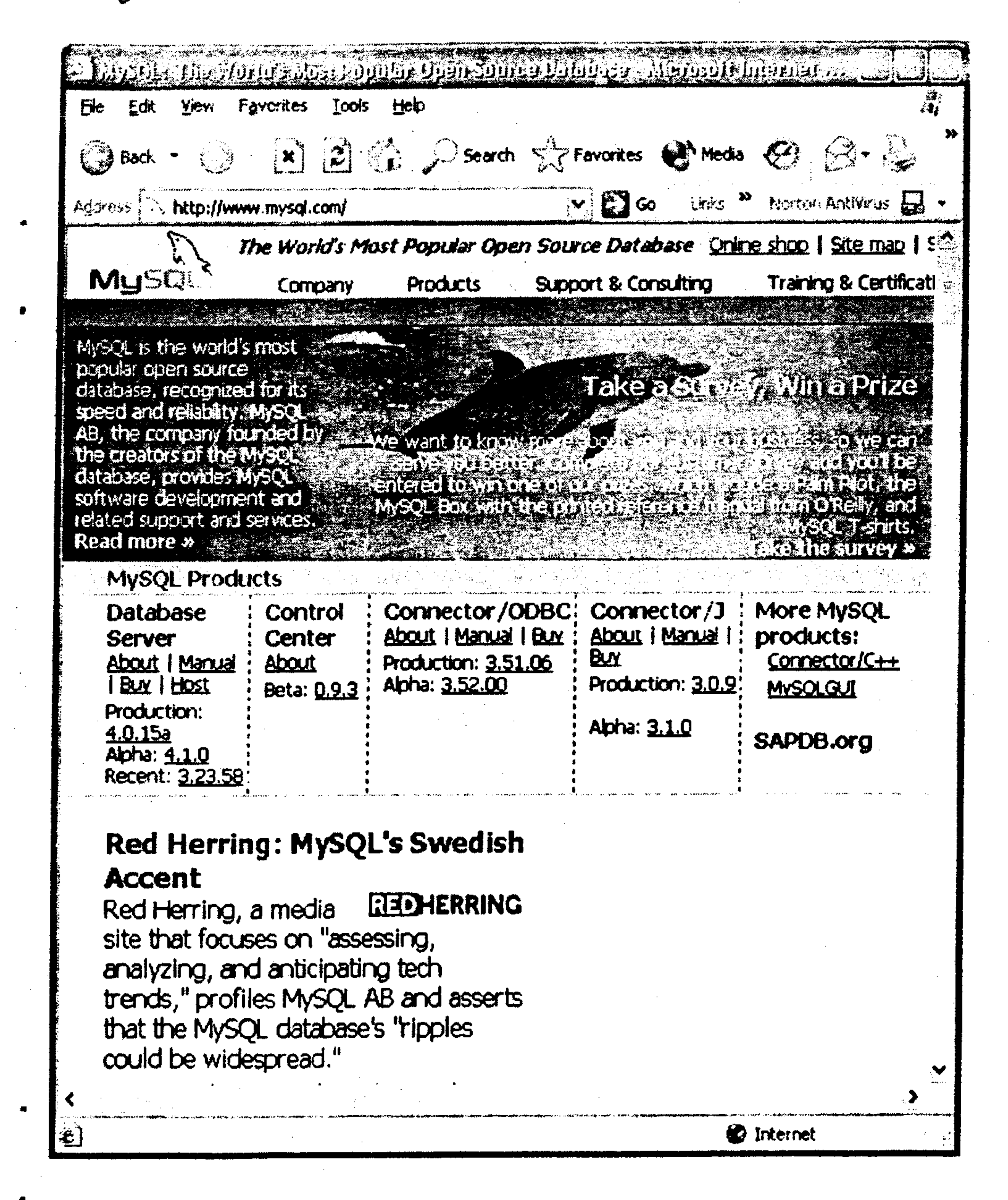
٥ - وحدة إدارة أقراص مرنة.

وبما أن MySQL من البرمجيات المفتوحة المصدر Open Source فيمكن الحصول عليها مجاناً دون اى اتفاقيات للترخيص..

ويمكن لأى مستخدم الحصول عليها من خلال الموقع التالى على الإنترنت :

http://www.mysql.com

- وعند الدخول لهذا الموقع تظهر النافذة الخاصة به كما يلى :


File Edit View Favorites Tools Help
Back
Search
Favorites
Media
Address http://www.mysql.com/
Go
Links
Norton AntiVirus
The World's Most Popular Open Source Database Online shop | Site map |
MySQL
Company
Products
Support & Consulting
Training & Certificati
MySQL is the world's most popular open source database, recognized for its speed and reliability. MySQL AB, the company founded by the creators of the MySQL database, provides MySQL software development and related support and services.
Read more »
Take a Survey, Win a Prize
MySQL T-shirts.
Take the survey »
MySQL Products
Database Server
About | Manual | Buy | Host
Production: 4.0.15a
Alpha: 4.1.0
Recent: 3.23.58
Control Center
About
Beta: 0.9.3
Connector/ODBC
About | Manual | Buy
Production: 3.51.06
Alpha: 3.52.00
Connector/J
About | Manual | Buy
Production: 3.0.9
Alpha: 3.1.0
More MySQL products:
Connector/C++
MySQLGUI
SAPDB.org
Red Herring: MySQL's Swedish Accent
Red Herring, a media site that focuses on "assessing, analyzing, and anticipating tech trends," profiles MySQL AB and asserts that the MySQL database's "ripples could be widespread."
REDHERRING
Internet

MR

* وفيما يلي بعض محتويات الموقع عن البرنامج وأنواعه:

" **http://www.mysql.com/products/mysql/index.html** "

There are four versions of the database server available:

- **MySQL Standard** includes the standard MySQL storage engines and the InnoDB storage engine. InnoDB is a transaction-safe, ACID-compliant storage engine with commit, rollback, crash recovery and row-level locking capabilities. This version is for users who want the high-performance MySQL database with full transaction support. MySQL Standard is licensed under the GPL. **MySQL Pro** is the commercially-licensed version of the server with the same feature-set.
- **MySQL Max** is for the user who wants early access to new features. This version includes the standard MySQL storage engines, the InnoDB storage engine, and other extras like the Berkeley database (BDB) storage engine, and support for splitting tables across multiple files to avoid operating system file size limitations. In future releases, MySQL Max will include more cutting-edge features.
- **MySQL Pro** is the commercially licensed version of the **MySQL Standard** database server, including InnoDB support.
- **MySQL Classic** only includes the standard MySQL storage engines, differing from MySQL Pro and MySQL Standard only by the omission of the InnoDB storage engine. It is only available under a commercial license.



* وفى صفحة التحميل Download بالموقع يمكنك مشاهدة التالى :

" **http://www.mysql.com/downloads/index.html** "

- **Mirrors** -- for faster downloads, use our download mirrors. Choose your closest mirror from here.
- **MySQL database server & standard clients:**
    - MySQL 4.0 -- Production release (recommended)
    - MySQL 4.1 -- Alpha release (use this for new development)
    - MySQL 5.0 -- Development tree (use this for previewing and testing new features)
    - MySQL 3.23 -- Older production release
    - Older releases -- older releases (only recommended for special needs)
    - Snapshots -- source code snapshots of the development trees
- **Graphical clients** -- different GUI interfaces to administer MySQL and data
    - MySQL Control Center
    - MySQLGUI (no longer under development)
- **Application Programming Interfaces (APIs)**
    - **Official APIs:**
        - The C API is included with the server, above.
        - Connector/ODBC - MySQL ODBC driver

        - Connector/ODBC 3.52 -- development release
        - Connector/ODBC 3.51 -- production release
        - Connector/ODBC 2.50 -- old release
    - MySQL Connector/J -- for connecting to MySQL from Java
        - MySQL Connector/J 3.1 -- development release
        - MySQL Connector/J 3.0 -- production release
        - MySQL Connector/J 2.0 -- old release
    - MySQL++ -- the official MySQL C++ API
  - **Contributed APIs:**
    - DBI -- for connecting to MySQL from Perl
    - Ruby -- for connecting to MySQL from Ruby
    - Python -- for connecting to MySQL from Python
    - .NET -- for connecting to MySQL from .NET
    - Ch -- for connecting to MySQL from Ch (C/C++ interpreter)

- **MySQL Logos** -- logos for use on your website

- **Other downloads:**
  - Linux
  - FreeBSD
  - NetBSD
  - Solaris
  - SCO

  - Win32
  - NT
  - OS/2

- **Contributed software** -- applications/APIs developed by third parties
- **Certified Hardware** -- Dedicated platforms for MySQL databases, certified by MySQL AB
- **Certified Hosting** -- Managed hosting of MySQL databases, certified by MySQL AB

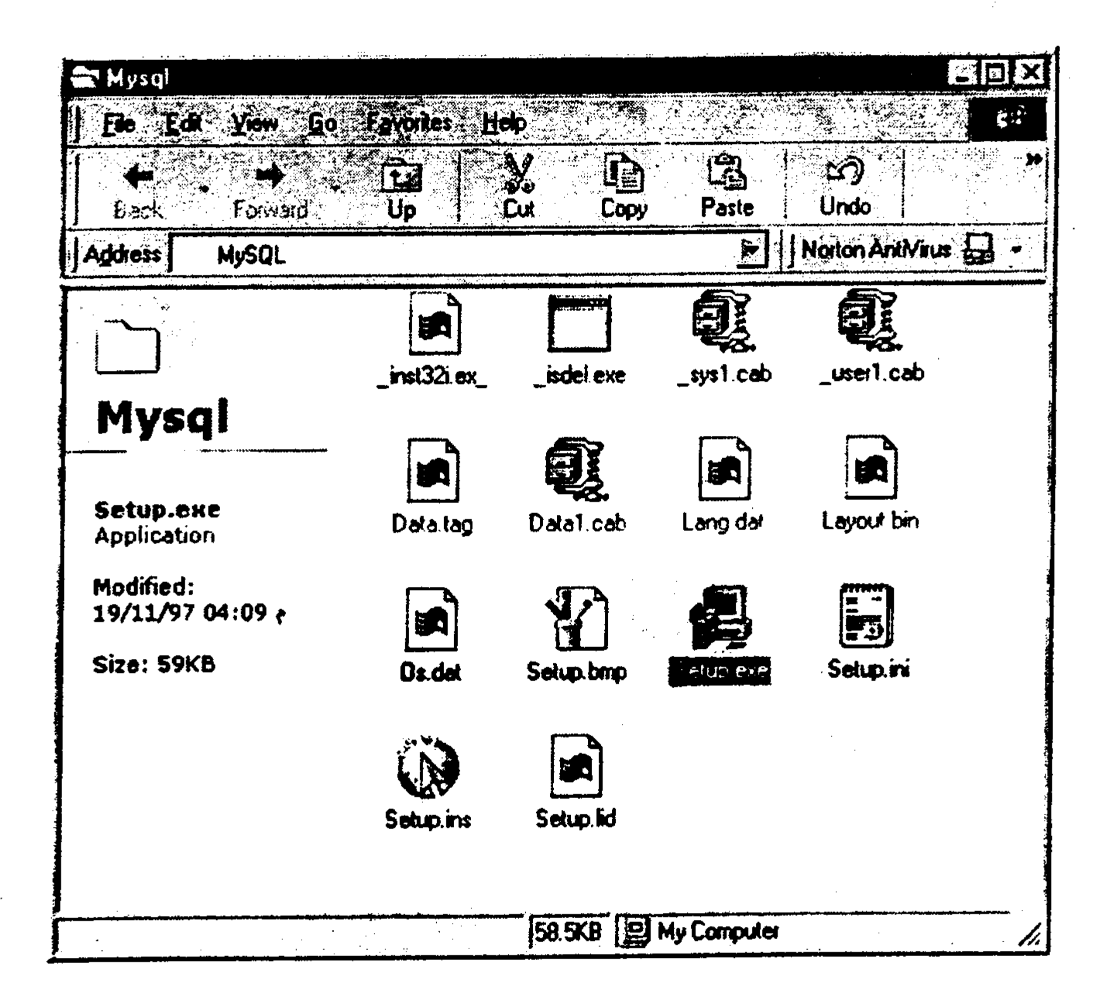
* وبعد الحصول على النسخة المصدرية Source الخاصة بقاعدة البيانـــات MySQL يمكنك اتباع الخطوات التالية لإعداد MySQL بنجاح :

ملاحظة : فى حالة وجود MySQL فى مجلد مضغوط بامتداد ZIP يمكنك استخدام برنامج Winzip لفك ضغط مجلد MySQL أولا ومـــن ثم متابعة الخطوات التالية ...

MR

*** خطوات الإعداد لـ MySQL :**

١- قم بفتح المجلد المسمى MySQL ليظهر أمامك الشكل التالى :



٢- من خلال النقر المزدوج DoubleClick على الرمز Icon المســمى Setup.exe يتم بدء إعداد MySQL وظهور شاشة تدل على بدء عملية الإعداد ...

انظر الشكل التالى:

mySQL

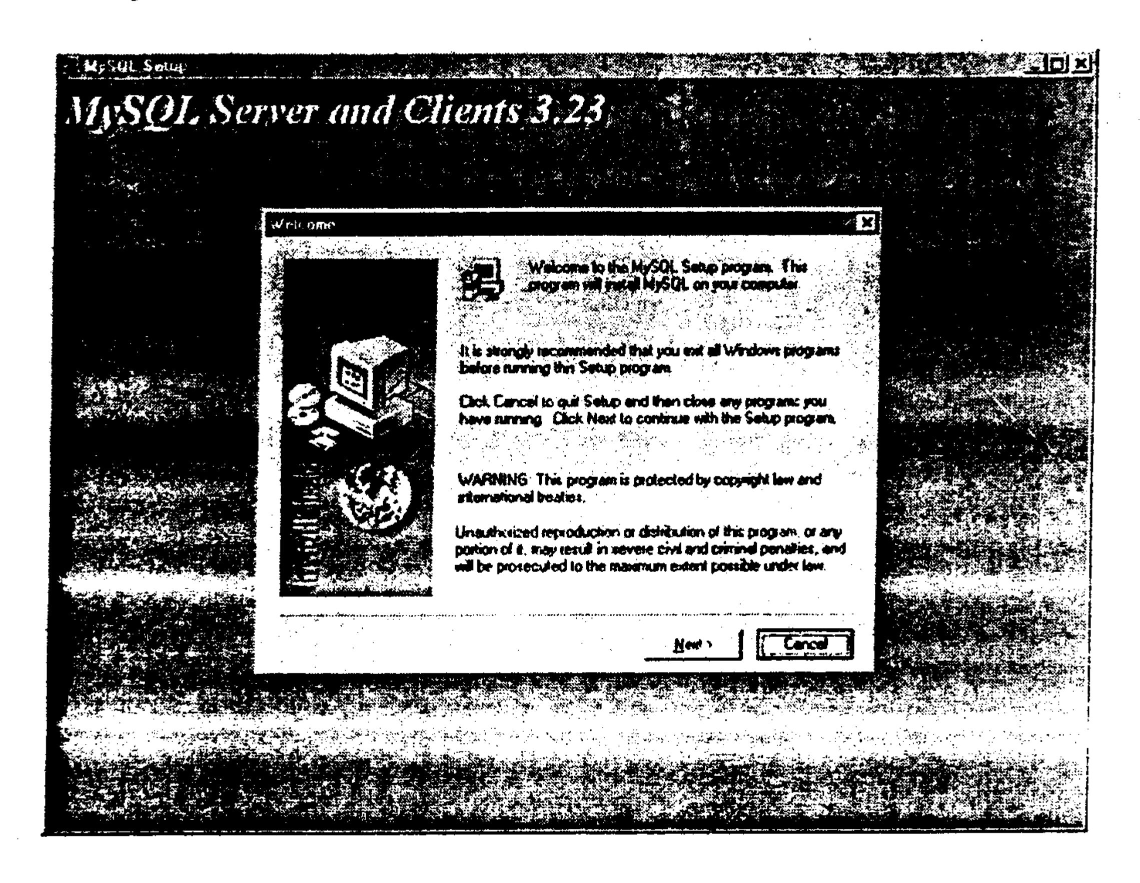
Setup



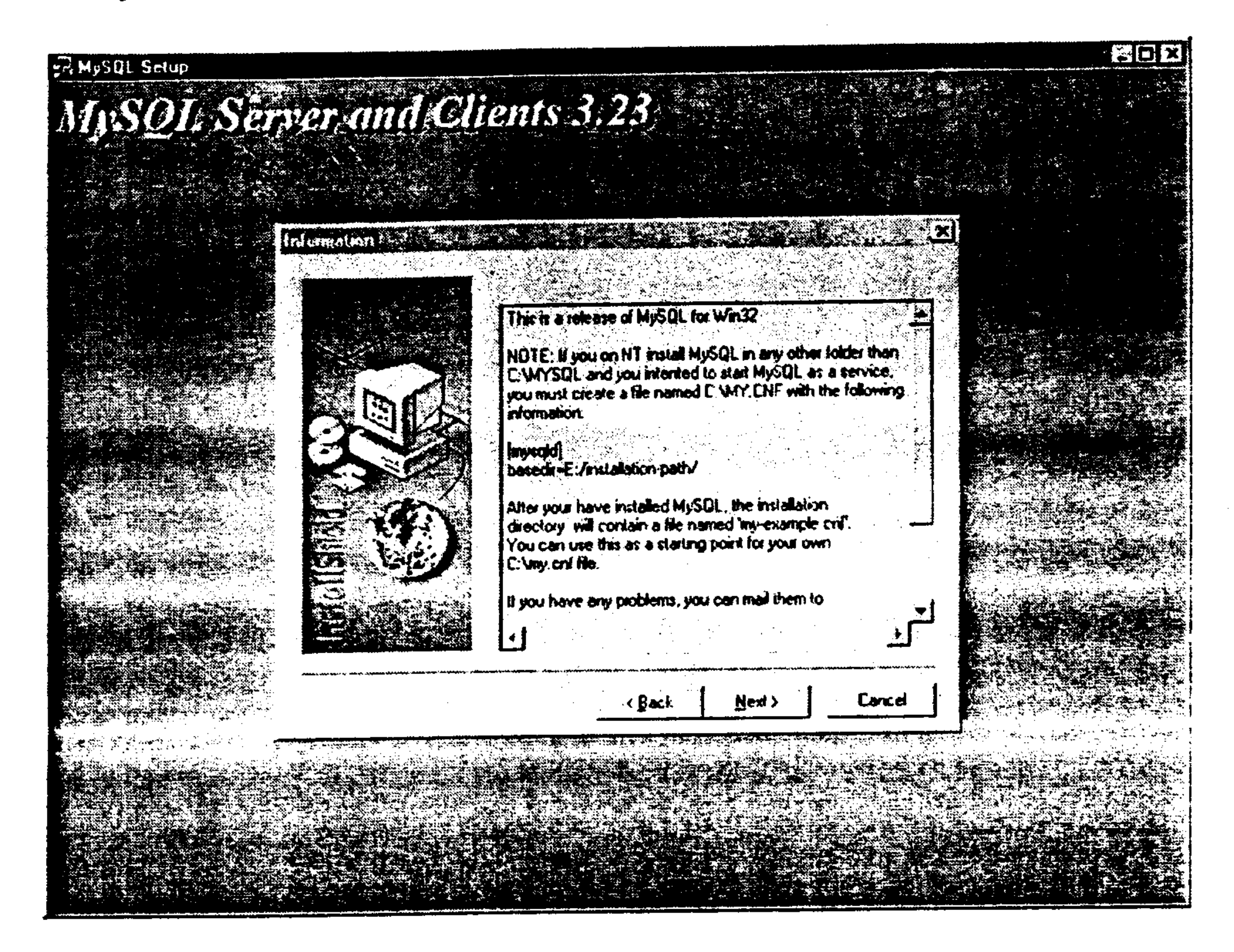
Setup is preparing the InstallShield® Wizard which will guide you through the application setup process. Please wait.

100 %

٣- بعد ذلك تظهر النافذة التالية فيتم النقر على زر Next (التالى).



٤- بعد ذلك تظهر النافذة التالية وبها بعض التعليمات فيتم النقر أيضا علـــى زر Next لمتابعة الإعداد :



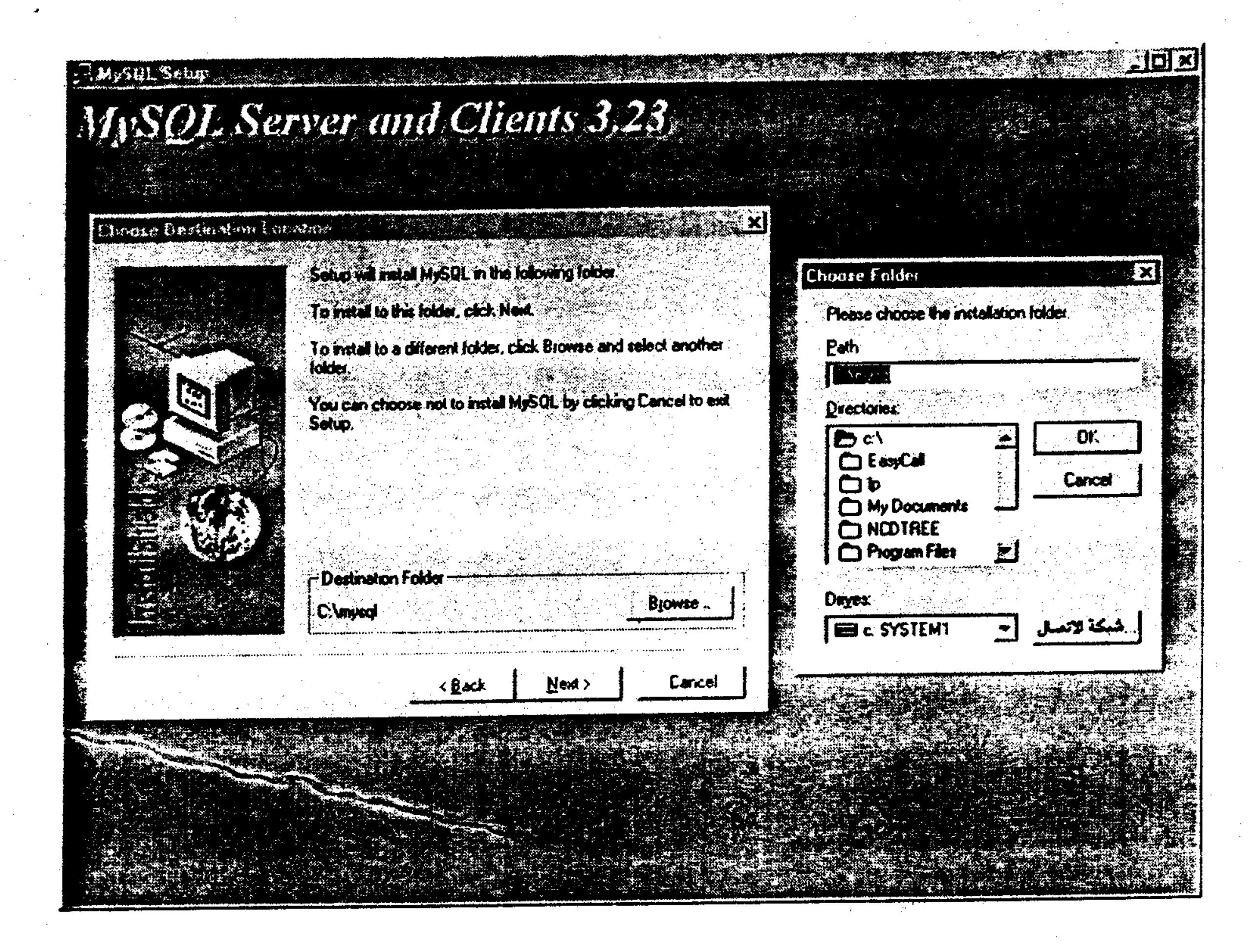
٥- فى النافذة التالية يتم تحديد المسار الذى سيتم فيه MySQL على القرص الصلب HDD .. وقد وضع البرنامج المسار

C:\mysql

كمسار افتراضى يمكنك تغيره، عن طريق النقر على الزر المسمى Browse الموجود جانب المسار السابق داخل الجزء المسمى :

Destination Folder

فيظهر مربع حوارى يتم من خلاله تحديد المســار المــراد إعــداد MySQL داخله .. ويمكنك ترك المسار الافتراضى واستكمال التحميل بالنقر على الــزر Next ..



وفى حالة الرغبة فى تغيير المسار يتم تغيير المسار فى النافذة المسماة Choose Folder ثم يتم النقر على الزر Ok لحفظ وإنشاء المسار الجديد فى حالة عدم وجوده من قبل ...

وفى الشكل التالى تم تغيير المسار إلى :

d:\server\mysql

وعند النقر على الزر Ok ظهرت النافذة بالشكل التالى..

Choose Destination Location

Setup will install MySQL in the following folder.

To install to this folder, click Next.

To install to a different folder, click Browse and select another folder.

You can choose not to install MySQL by clicking Cancel to exit Setup.

Destination Folder

d:\server\mysql

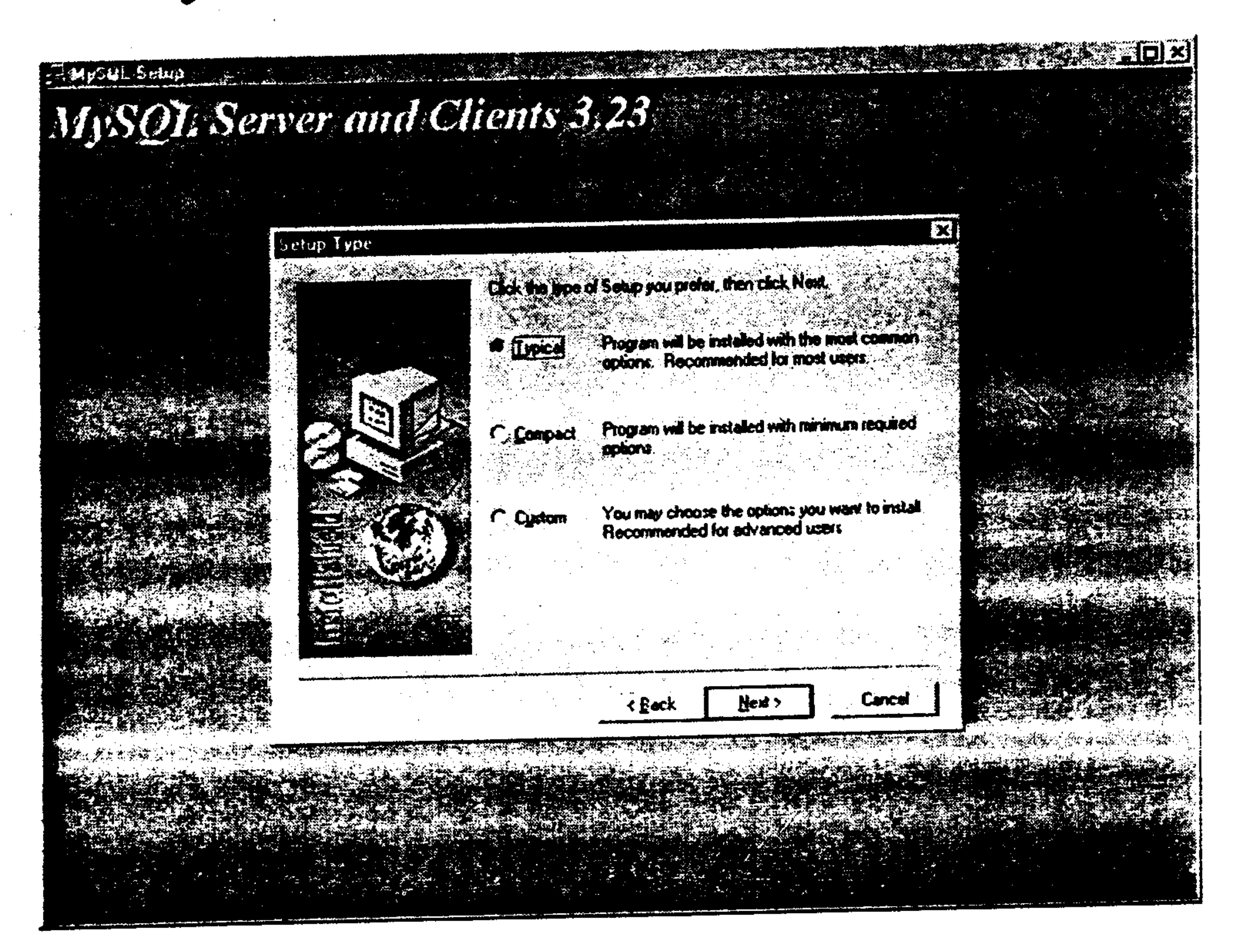
Browse...

< Back | Next > | Cancel

وللمتابعة يتم النقر على زر Next ..

٦- وبعد تحديد المسار فى الخطوة السابقة يتم فى هذه الخطوة تحديد نـــوع الإعداد للـــ MySQL ، وهناك ثلاث أنواع من الإعداد :

انظر الشكل التالى :



الأول **Typical**

يتم إعداد البرنامج بإضافة أغلب ملحقات البرنامج ...
وهو الاختيار الأفضل لأغلب المستخدمين.

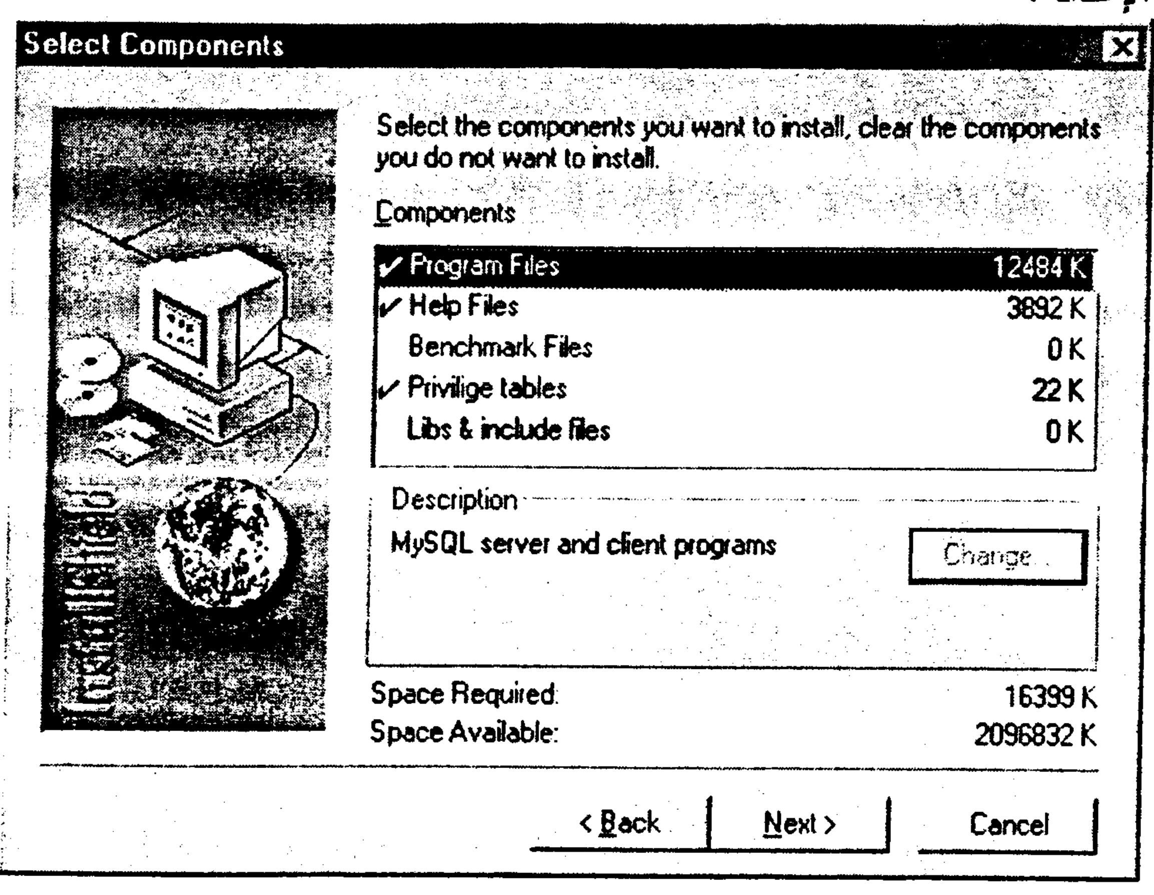
الثانى **Compact**

يتم إعداد البرنامج بإضافة الحد الأدنى من متطلبات التشغيل للبرنامج

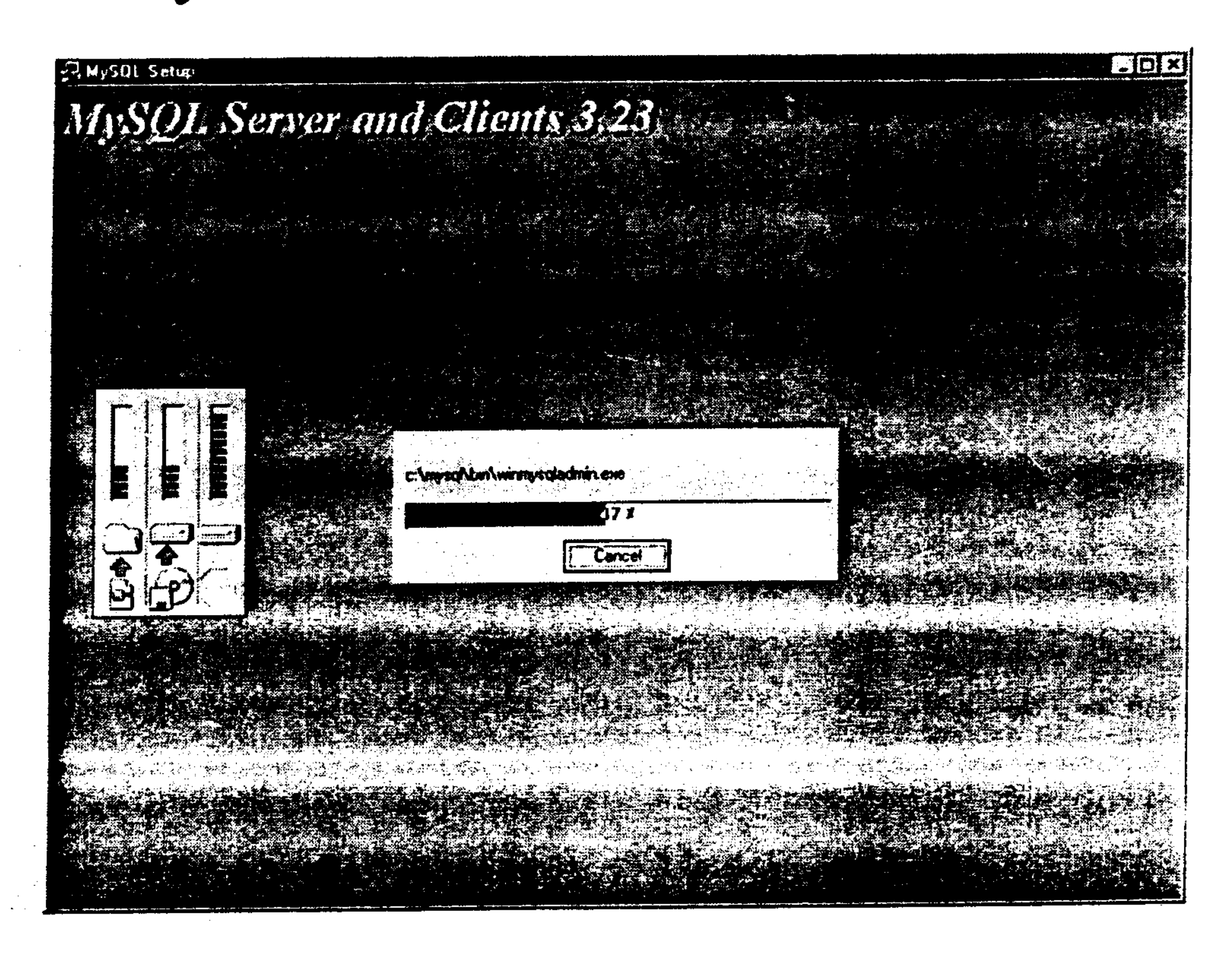
الثالث **Custom**

يمكن للمستخدم تحديد أو إلغاء تحديد الخيارات التى يريد إعدادها مع البرنامج.
وهو الاختيار الأفضل للمستخدمين المحترفين

وعند اختيار طريقة الإعداد المطلوبة (Typical أو Compact) يتم النقر على زر Next لبدء التحميل للـــ MySQL مباشرة ...
أما عند اختيار الطريقة Custom تظهر النافذة التالية قبل البدء فـــى عمليـــة الإعداد :



٧- يتم الانتظار حتى يتم استكمال الإعداد ... وبذلك يكون MySQL قد تم إعداده بنجاح داخل الدليل الذى تم تحديده من قبل المستخدم فى الخطـــوة رقم ٥.
انظر الشكل التالى :



* عند ذلك نكون قد انتهينا من عملية إعداد MySQL على الجهاز ... وبقى الآن أن نشرح كيفية تشغيل البرنامج داخل بيئة Windows ..

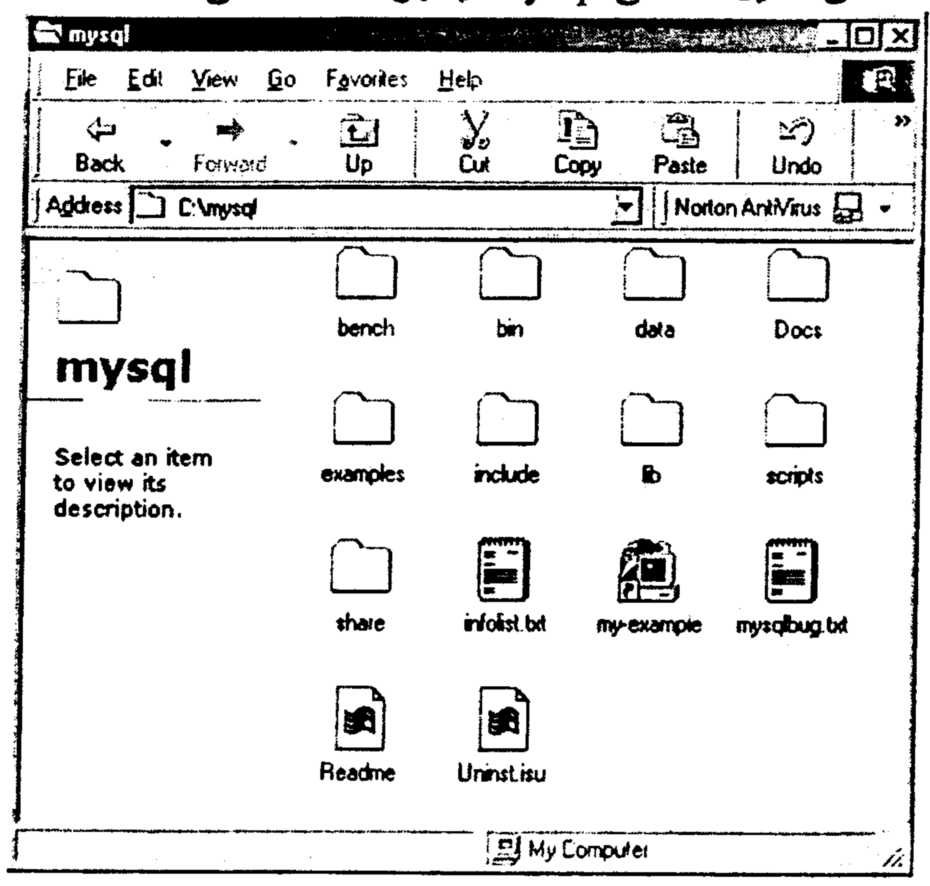
==========================

٤-٢ التشغيل :

بعد الانتهاء من إعداد MySQL كما سبق شرحه ... تـأتى الآن عمليـة تشغيل محرك MySQL من داخل بيئة Windows وذلك لنتمكن من التعامـل مع قاعدة البيانات MySQL وأوامرها ...

وللتشغيل يتم إتباع الخطوات التالية :

١- فتح الرمز المسمى My Computer من على سطح المكتب ...
٢- فتح محرك الأقراص \:C (بفرض انه لم يتم تغير المسار الافتراضى للـ MySQL أثناء عملية الإعداد) ..
٣- البحث عن الدليل المسمى mysql ليظهر الشكل التالى :



MR

٤- يتم فتح الدليل (المجلد) المسمى " **bin** " ليظهر الشكل التالى :

bin

File Edit View Go Favorites Help

Back Forward Up Cut Copy Paste Undo Delete

Address C:\mysql\bin

Norton AntiVirus

bin

Select an item to view its description.

isamchk.exe myisamchk.... myisamlog.exe myisampack... mysql.exe

mysqladmin.... mysqlbinlog.... mysqlc.exe mysqld.exe mysqld-nt.exe

mysqld-opt.exe mysqldump.... mysqlimport.... MySqlManag... mysqlshow.exe

mysqlshutdown.exe mysqlwatch.... pack_isam.exe perror.exe replace.exe

winmysqlad... winmysqlad... WINMYSQ...

11.0MB My Computer

٥- يتم النقر المزدوج على الرمز المسمى *winmysqladmin.exe* وذلك لبدء تشغيل محرك MySQL ...

عند ذلك يظهر الشكل التالى :

WinMySQLadmin 1.0

WinMySQLadmin Ver 1.0 for Win95/Win98/NT
Copyright (C) 1979-1999 TcX AB Monty Program KB Detron HB.
All rights reserved. See the file PUBLIC for licence information.
This software comes with ABSOLUTELY NO WARRANTY: see the file PUBLIC for details

Right Click for Menu options

Environment | Start Check | Server | my.ini Setup | Err File | Variables | Process | Databases | Report

Environment
Local Host Name
Local User Name
OS Platform
Local IP Address
Total Physical Memory

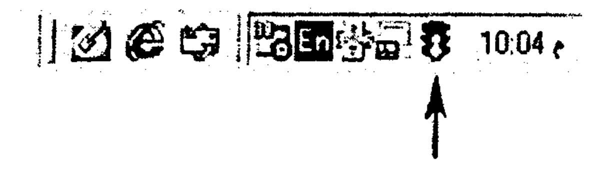
MyODBC
Not Found

Server
Server Info | Host Info | Open tables
Client Info | Protocol Info | Open files
Uptime | Threads running | Open streams
Slow queries | Opened tables | Questions

Powered by mySQL

Hide me | Stop Extended Server Status

وهذه النافذة تبقى مفتوحة لمدة ثوانى قليلة ثم تختفى ليظهر رمــز بشـــريط المهام Taskbar بجانب الساعة يدل على أن MySQL يعمل، والشكل التـــالى يوضح شكل هذا الرمز :



بهذا يكون MySQL جاهز لبدء التعامل معه ...

بالطبع سيتم فى كل مرة إعادة تشغيل الجهاز تنفيذ الخطوات السابقة لفتح محرك MySQL .. ولتجنب ذلك يمكن إنشاء اختصار Shortcut على سطح المكتب Desktop وذلك ليتم التشغيل بمجرد النقر المزدوج على هذا الاختصار ..

ولتنفيذ ذلك يتم إتباع الخطوات التالية :

١- فتح My Computer ومن ثم فتح محرك الأقراص :C ، ثم فتح الدليل mysql ثم فتح الدليل الفرعى المسمى bin، وذلك ليكون الدليل الحالى هو :
C:\mysql\bin
(وقد سبق شرح ذلك فى الجزء السابق)

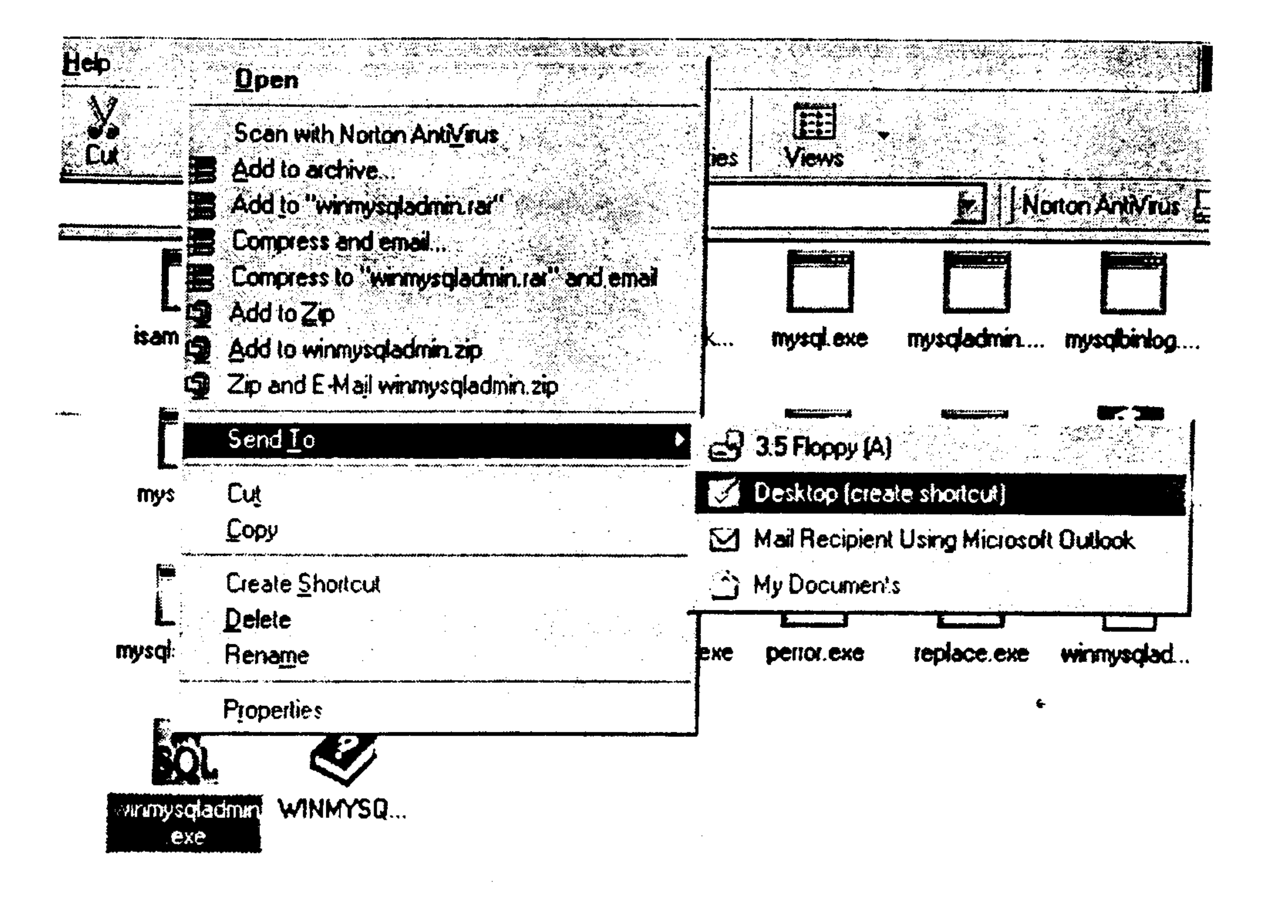
٢- النقر بالزر الأيمن على الرمز *winmysqladmin.exe* لتظهر قائمة مختصرة.

٣- يتم الذهاب للقائمة الفرعية Send To واختيار الأمر :
Desktop (create shortcut)

بذلك يتم إنشاء اختصار على سطح المكتب بمجرد النقر المزدوج عليه يتم تشغيل محرك البرنامج مباشرة.

انظر الشكل التالى :





ملاحظة : فى حالة الرغبة فى تشغيل Mysql تلقائيا فى كل مرة يتم فيها تشغيل الـ Windows يمكن وضع اختصار الرمز winmysqlmyadmin.exe فى القائمة StartUp الموجودة فى قائمة Start>Programs

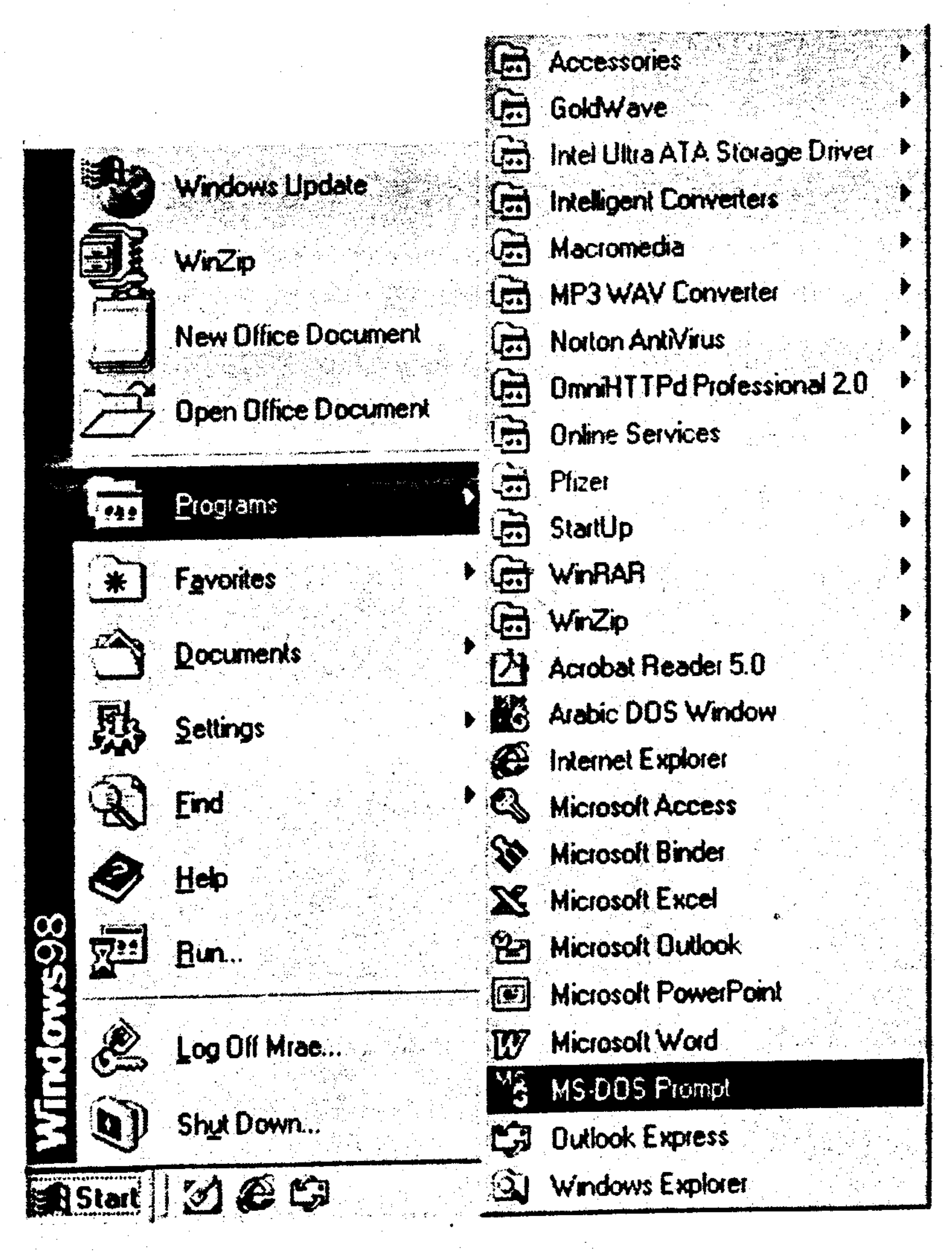
بعد أن تم تشغيل محرك *MySQL* سيتم الآن شرح كيفية الدخول لعلامة نظام *MySQL* وذلك لتنفيذ الأوامر والعمليات من خلالها ... (وسيكون ذلك فى بيئة الـ*DOS*) ..

ولتنفيذ ذلك يتم إتباع الخطوات التالية :

MR

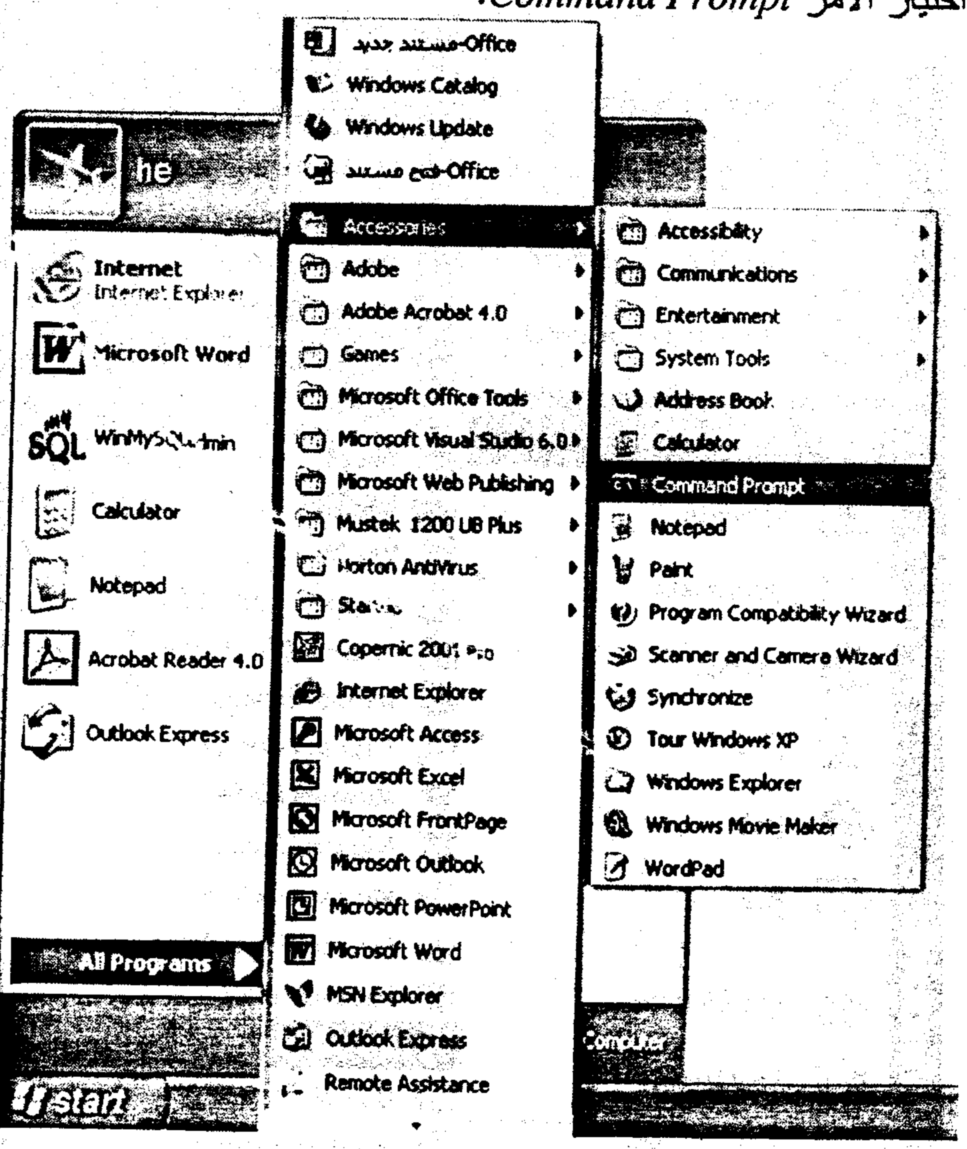
١- فتح قائمة ابدأ *Start*.
٢- فتح القائمة الفرعية *Programs*.
٣- تشغيل الرمز المسمى *MS-DOS Prompt*.

انظر الشكل التالى :



ملاحظـــة : بـــالطبع خطـــوات فـــتح *MS-DOS Prompt* الســـابقة خاصـــة بالــــ *Windows98* أما فى الـــ *WindowsXP* فيمكن تنفيذ ذلك من خلال :

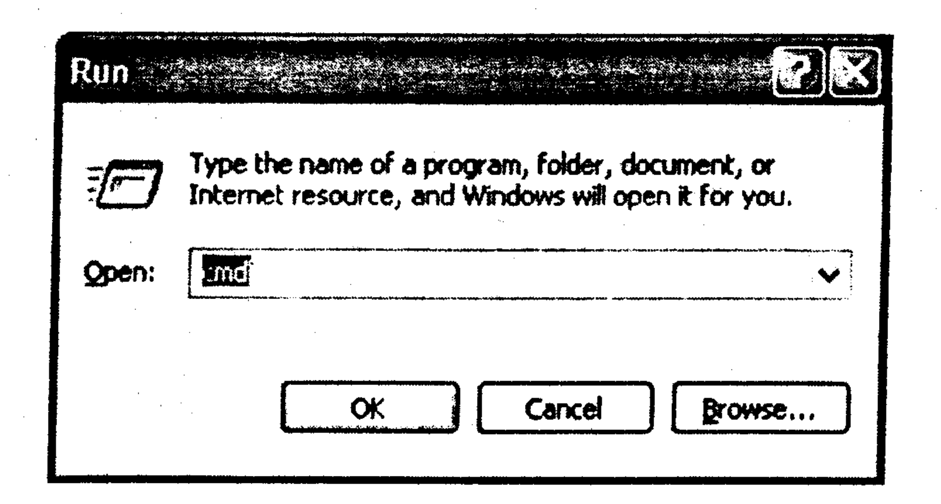
- فتح قائمة *Start*
- فتح القائمة الفرعية *All Programs*
- فتح القائمة الفرعية *Accessories*
- واختيار الأمر *Command Prompt*.



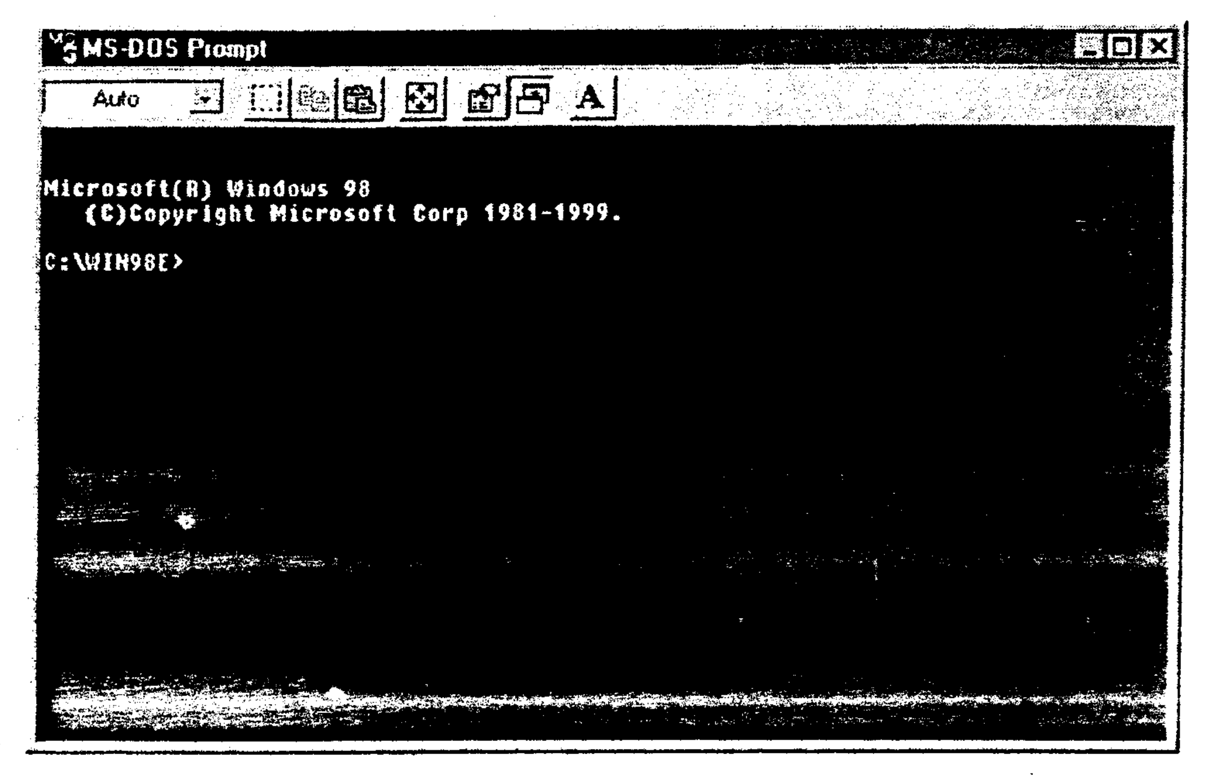
MR

* **ويمكن أيضا التنفيذ من خلال :**

- فتح قائمة *Start*
- اختار الأمر *...Run*
- كتابة الأمـــر التـــالى (*cmd*) ثـــم النقـــر علـــى *Enter* ليـــتم فـــتح نافـــذة الـــ *Dos Prompt*.



٤- تظهر النافذة التالية:



فيتم كتابة الأوامر التالية للخروج من دليل الـ *Windows* والانتقال إلى دليل الـ *MySQL* وتشغيله :

```
C:\windows> cd \          <Enter>
C:\>cd mysql\bin          <Enter>
C:\mysql\bin>mysql        <Enter>
```

أمر التشغيل

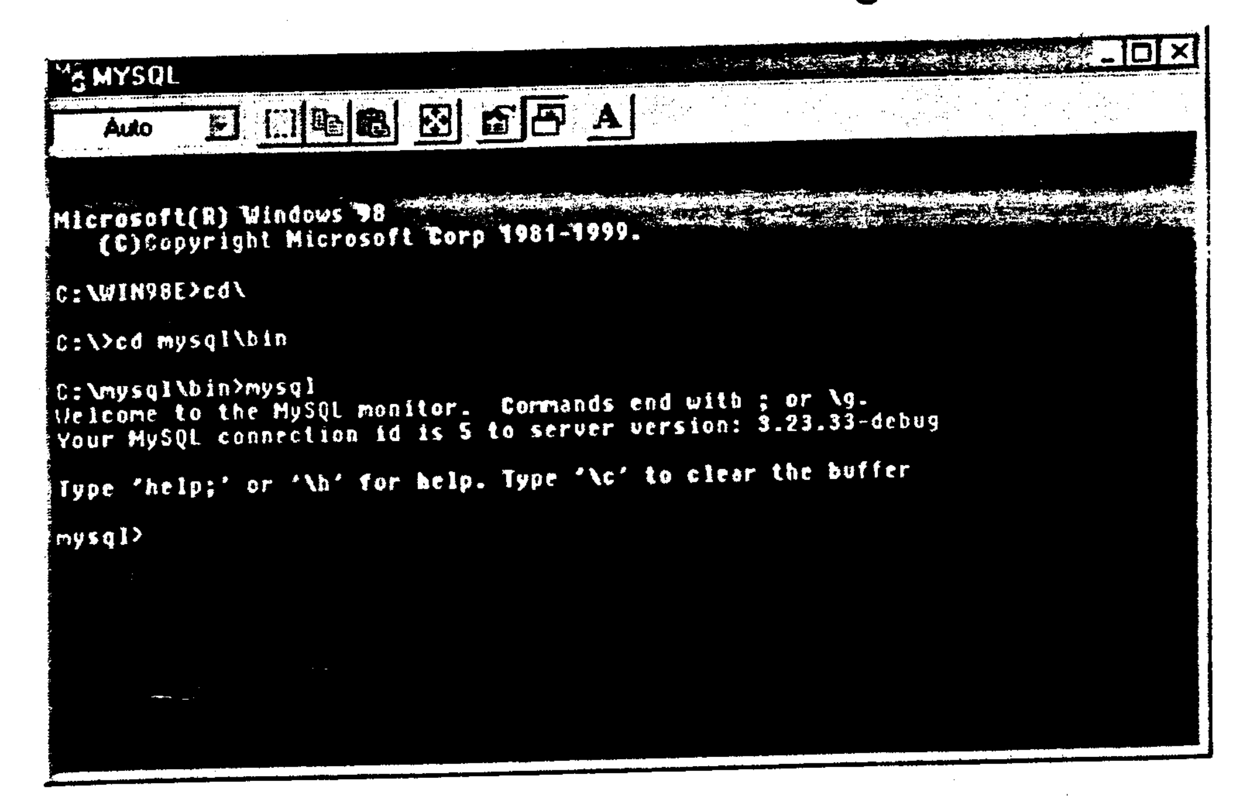
عند ذلك يظهر الآتى:

```
Welcome to the MySQL monitor.  Commands end with ; or \g .
Your MySQL connection id is 3 to server version: 3.23.33-debug

Type 'help; ' or '\h' for help. Type '\c' to clear the buffer

mysql>
```

انظر الشكل التالى :



* بهذا تظهر علامة *MySQL* ليكون جاهز لاستقبال الأوامر المختلفة من المستخدم.

٤-٣ الخروج من MySQL والإغلاق النهائى له ..

أ) الخروج من نافذة MySQL :

بعد تنفيذ الأوامر المطلوبة والرغبة فى الخروج من علامة البرنـــامج يـــتم كتابة الأمر Exit أمام علامة الـــ MySQL كالتالى :

mysql> exit;

فيكتب على الشاشة الأتى :

```
mysql> exit;
Bye

C:\mysql\bin>
```

ثم يتم إغلاق نافذة *MS-DOS Prompt* ..

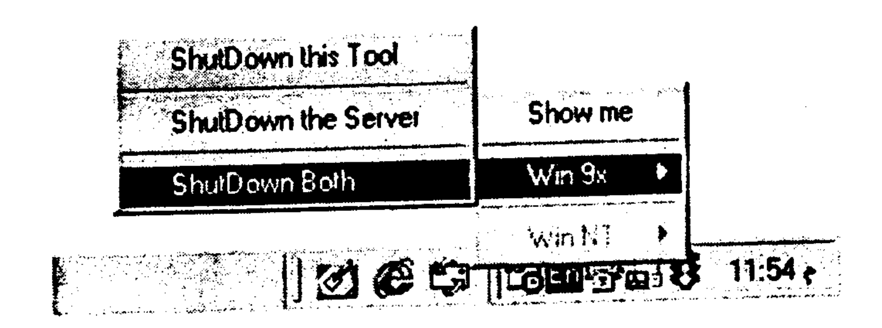
ب) إغلاق النهائى لـــ *MySQL* :

عند الرغبة فى إنهاء محرك *MySQL* ... (وفى هذه الحالة عند الرغبـــة فـــى إعادة تشغيله سيتم إتباع خطوات التشغيل السابق شرحها فى ثانيا).. يتم إتباع الخطوات التالية:

١- الذهاب لرمز البرنامج على شريط المهام الموجود بجانب الساعة والنقر عليه بالفأرة نقرة واحدة ، لتظهر قائمة هذا الرمز.
٢- يتم الذهاب للقائمة الفرعية *Win 9x* واختيار الأمر

ShutDown both

انظر الشكل التالى :



صورة *shutdown mysql.bmp*

* وتوجد بقائمة *Win 9x* ثلاث أوامر وهى :

1- ShutDown this Tool

وهذا الأمر خاص بإغلاق الأداة الخاصة بالبرنامج فقط، والموجودة بجانب الساعة، وفى هذه الحالة ستختفى الأداة من جانب الساعة بشريط المهام " *System Tray* " ولكنها لن تؤثر على التعامل مع *MySQL*.

2- ShutDown this Server

هذا الأمر خاص بإغلاق *Server* الـــ *MySQL*، وفى هـــذه الحالـــة لـــن نتمكن من التعامل مع *MySQL* حتى يتم إعادة تشغيله مرة أخرى.

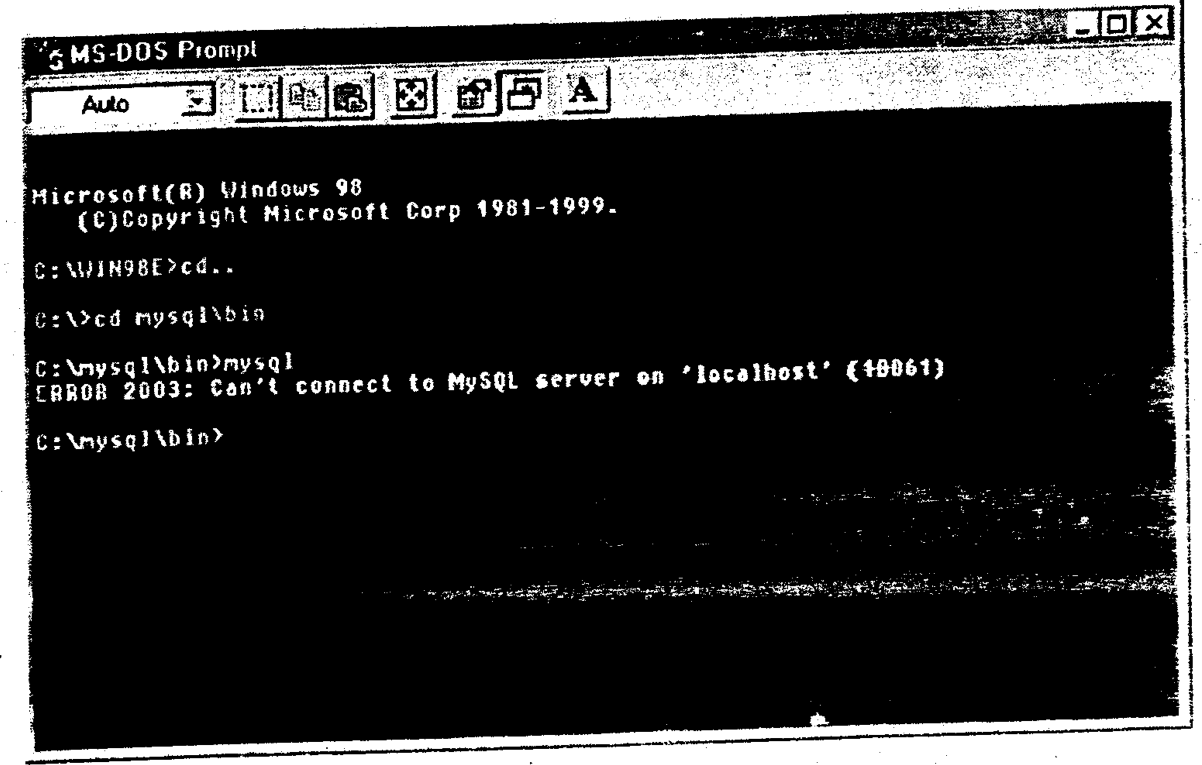
ويلاحظ تغير لون الرمز الموجود بجوار الساعة من اللون الاخضر للـــون الأحمر للدلالة على ان الـــ *Server* قد تم إغلاقه.

وعند محاولة تشغيل *MySQL* من *MS-DOS Prompt* تظهر الرســـالة التالية :

```
C:\mysql\bin>mysql
ERROR 2003: Can't connect to MySQL server on 'localhost' (10061)

C:\mysql\bin>
```

انظر الشكل التالى :



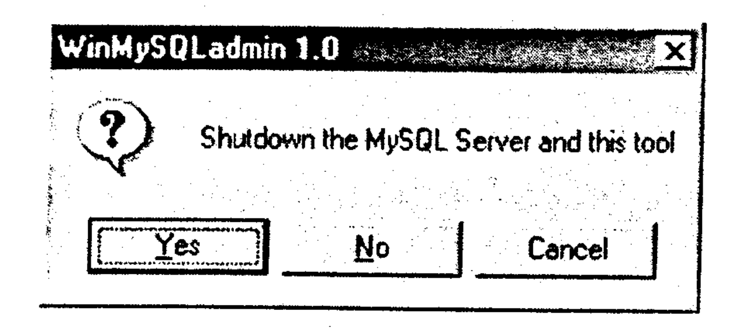
- ولتشغيل الـ *Server* مرة أخرى يتم فتح القائمة *Win 9x* لتجد ان الأمر *ShutDown this Server* تحول إلى *Start the Server* .. وبالنقر عليه مرة واحدة بزر الفأرة يظهر مربع حوارى للتأكد من العملية فيتم النقر على الزر *Yes* ليبدأ الـ *Server* فى العمل مرة أخرى.

3- ShutDown Both

لإنهاء كلاً من الـ *Tool* والـ *Server* فى خطوة واحدة، وهذا هو الأمر المفضل فى حالة الرغبة فى إنهاء التعامل مع الـ *MySQL* بالكلية.

وفى كل مرة يتم تنفيذ أحد الأوامر السابقة يظهر مربع حوارى للتأكـد من تنفيذ الأمر، فيتم النقر على الزر *Yes* لاستكمال التنفيذ ...

والشكل التالى يوضح المربع الحوارى الذى يظهر مع الأمـر الأخيـر *ShutDown Both*



٤-٤ إزالة *MySQL* بالكلية من الـ *Windows* :

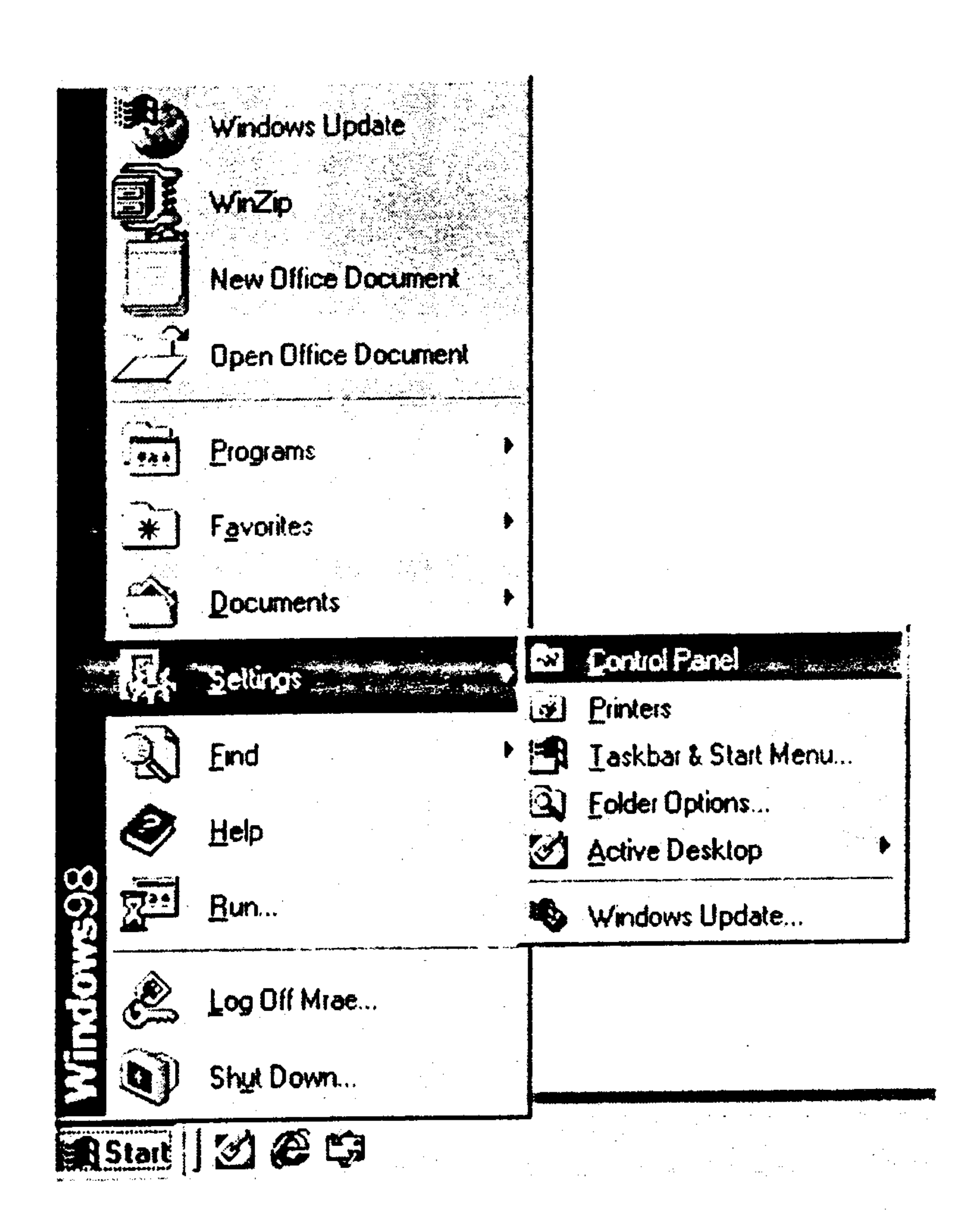
عند الرغبة فى إزالة البرنامج بالكلية من نظام التشغيل بغرض الإزالة فقط او إضافة نسخة أحدث أو غير ذلك من دواعى إلغاء البرامج، يتم اتباع الخطوات التالية لتنفيذ عملية الإزالة:

١- إغلاق *MySQL* (كما سبق شرحه)
٢- فتح لوحة التحكم *Control Panel* عن طريق :

Start > Setting > Control Panel

لتظهر نافذة لوحة التحكم ...

انظر الشكل التالى :


Windows Update
WinZip
New Office Document
Open Office Document
Programs
Favorites
Documents
Settings
Find
Help
Run...
Log Off Mrae...
Shut Down...
Windows98
Start
Control Panel
Printers
Taskbar & Start Menu...
Folder Options...
Active Desktop
Windows Update...

MR



٣- يتم اختيار الرمز *Add/Remove Programs* (إضافة وإزالة البرامج) وفتحه ..

لتظهر النافذة التالية :

Add/Remove Programs Properties

Install/Uninstall | Windows Setup | Startup Disk

To install a new program from a floppy disk or CD-ROM drive, click Install.

Install...

The following software can be automatically removed by Windows. To remove a program or to modify its installed components, select it from the list and click Add/Remove.

Microsoft Office 97, Professional Edition
MP3 WAV Converter 2.65
MySQL Server and Clients 3.23
Norton AntiVirus 2002
NVIDIA Windows 95/98/ME Display Drivers
OmniHTTPd Professional v2.10
PCI Audio Driver
UPDATE YOUR SYSTEM
WinRAR archiver

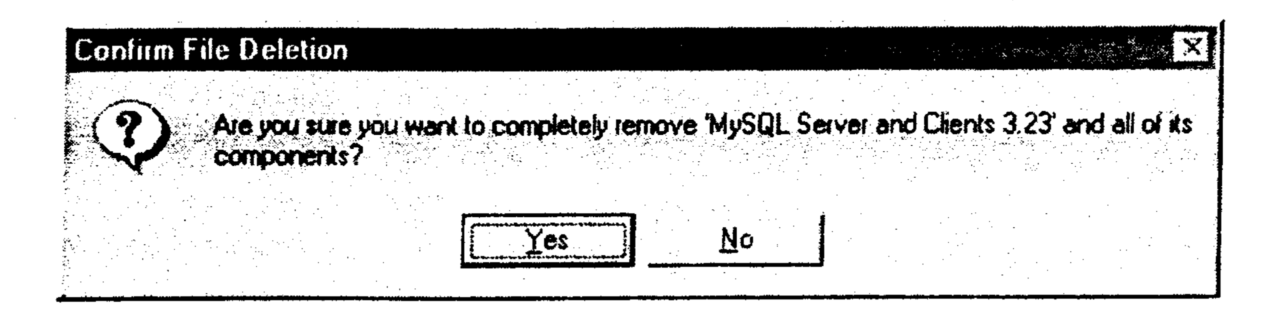
Add/Remove...

OK | Cancel | Apply

يتم اختيار اسم البرنامج :

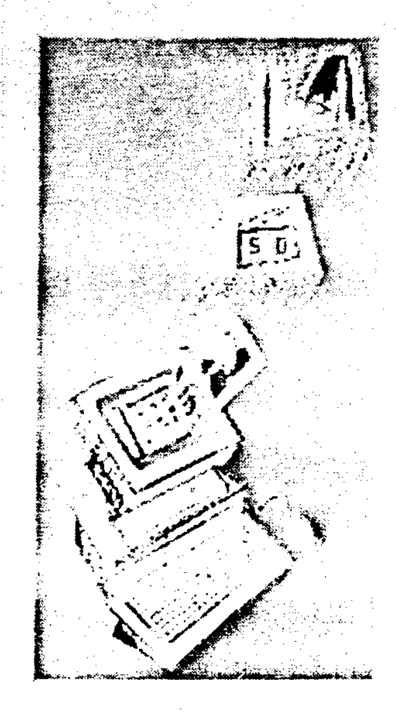
MySQL Server and Clients 3.23

٤- النقر على الزر المسمى *Add/Remove*، بذلك يظهر المربع الحوارى التالى فيتم النقر على الزر *Yes*.



٥- تظهر النافذة التالية ... ويتم الانتظار حتى ينتهى الإلغاء للبرنامج ومن ثم يتم النقر على زر *OK* ..

Remove Programs From Your Computer



unInstallShield will remove the software 'MySQL Server and Clients 3.23' from your computer. Please wait while each of the following components is removed...

- ✔ Shared program files...
- ✔ Standard program files...
- Folder items...
- Program folders...
- Program directories...
- Program registry entries...

بهذا يكون قد تم إلغاء *MySQL* بشكل كامل مـن الـ *Windows* ...
وعند الرغبة فى تشغيله مرة أخرى يجب إعادة الإعداد من جديد.

==================

mySQL

التعامل التطبيقى مع MySQL

الفصل الخامس

التعامل مع
قواعد البيانات وجداولها

يتناول هذا الفصل المحتويات الآتية :

- التعامل مع قواعد البيانات
- التعامل مع الجـــداول
- مثـــال تطـبيقى

٥-١ التعامل مع قواعد البيانات

٥-١-١ إنشاء قاعدة بيانات جديدة:

الأمــر:

<CREATE DATABASE>

الصيــغة:

create database <Database name>

الصيغة بالعربية للتوضيح:

create database < اسم قاعدة البيانات الجديدة >

مثال:

mysql > ***create database*** refaat ;

وعند التنفيذ سيظهر على الشاشة شكل مشابه للشكل التالى :

```
C:\mysql\bin>mysql
Welcome to the MySQL monitor.  Commands with end  ; or  \g.
Your MySQL connection id is 4 to server version: 3.23.33

Type 'help; ' or  '\h' for help. Type '\c' to clear the buffer

mysql> CREATE DATABASE refaat;
Query OK, 1 row affected (0.00 sec)

mysql>
```

٥-١-٢ عرض أسماء قواعد البيانات الموجودة أو السابق إنشائها:

الأمــر:

<SHOW DATABASES>

الصيــغة :

show databases

مثال:

Mysql > ***show databases ;***

وعند التنفيذ سيظهر على الشاشة شكل مشابه للشكل التالى :

```
mysql> SHOW DATABASES;
+----------+
| Database |
+----------+
| mysql    |
| refaat   |
| test     |
+----------+
3 rows in set (0.00 sec)

mysql>
```


MR

٥-١-٣ فتح قاعدة بيانات تم إنشائها من قبل :

الأمــر:

< USE>

الصيــغة:

use <Database name>

الصيغة بالعربية للتوضيح :

use < اسم قاعدة البيانات المراد فتحها >

مثال:

mysql > ***use*** refaat ;

وعند التنفيذ سيظهر على الشاشة شكل مشابه للشكل التالى :

```
mysql> USE refaat ;
Database changed
mysql>
```

*** لاحظ ظهور العبارة:**

```
Database changed
```

وهى تدل على أن فتح قاعدة البيانات قد تم بنجاح.

٥-١-٤ حذف قاعدة بيانات بأكملها:

الأمــر:

< DROP DATABASE>

الصيــغة:

drop database <Database name>

الصيغة بالعربية للتوضيح :

drop database < اسم قاعدة البيانات المراد حذفها >

مثال:

mysql > **drop database** refaat ;

وعند التنفيذ سيظهر على الشاشة شكل مشابه للشكل التالى :

```
mysql> DROP DATABASE refaat ;
Query OK, 3 rows affected (0.06 sec)

mysql>
```

وبالطبع عند تنفيذ الأمر SHOW DATABASE لن يظهر اسم قاعدة البيانات refaat بين أسماء قواعد البيانات الموجودة .. وهذا يدل على أنها قد حذفت بالفعل.

ملاحظة :

هذا الأمر يحذف قاعدة البيانات نهائياً بما تحتويه من جداول وبيانات، فيجب التأكد جيداً والحذر عند استخدامه.

- راجع أنواع الحقول المختلفة فى الفصول السابقة صفحة ٤٨ ...

٥-٢ التعامل مع الجداول

٥-٢-١ إنشاء الجداول :

الأمـــر :

<CREATE TABLE>

الصيـــغة :

create table <Table name> <(Columns)>

الصيغة بالعربية للتوضيح :

create table <اسم الجدول>
<أسماء الحقول وتوصيفها ويفصل بين (الحقول بفاصلة)>

مثال:

```
mysql> create table tbref1 (name char(15) ,
         address char(40), salary float(4.2) ) ;
```


MR

وهو عبارة عن إنشاء جدول باسم " tbrefl " ، يتكون من ثلاثـــة حقـــول ، الحقل الأول الاسم ويسمى name ونوعه حرفي وسعته ١٥ حرفاً، والحقل الثانى العنوان ويسمى address ونوعه حرفي وسعته ٤٠ حرفاً، والحقل الثالث المرتب ويسمى salary ونوعه عشرى (ذو نقطة عائمة).

وعند التنفيذ سيظهر على الشاشة شكل مشابه للشكل التالى :

```
mysql> CREATE TABLE  tbrefl (
    ->name char(15),
    ->address char(40),
    ->salary float(4,2));
ERROR 1046: No Database Selected
mysql>
```

**** لاحظ ظهور العبارة :**

```
ERROR 1046: No Database Selected
```

وهى تدل على تعذر إنشاء الجدول وذلك للسبب التالى : "عدم وجود قاعدة بيانات مفتوحة حاليا ".

** وللتخلص من هذا الخطأ يجب إجراء أحد أمرين:

الأول : إنشاء قاعدة بيانات جديدة قبل تنفيذ هذا الأمر ..
عن طريق الأمر : Create Database <database name>
ولكن فى حالة وجود قاعدة بيانات سابقة الإنشاء ويراد إنشاء هذا الجدول بهـا

يتم تنفيذ الأمر الثانى.

الثانى : فتح قاعدة بيانات موجودة عن طريق الأمر التالى :
use < database name >

عند ذلك لن تظهر رسالة الخطأ ..

وسيكون ناتج تنفيذ الأمر مشابهة للشكل التالى :

```
mysql> CREATE TABLE  tbref1 (
    ->name char(15),
    ->address char(40),
    ->salary float(4,2));
Query OK, 0 rows affected (0.06 sec)

mysql>
```

* والعبارة التالية تدل على أن عملية إنشاء الجدول قد تمت بنجاح :

```
Query OK, 0 rows affected (0.06 sec)
```

٥-٢-٢ عرض أسماء الجداول الموجودة بقاعدة البيانات الحالية:

الأمــر :

<SHOW TABLES>

الصيــغة :

show tables

مثال :

mysql > ***show tables;***

* وعند التنفيذ سيظهر على الشاشة شكل مشابه للشكل التالى:

```
mysql> SHOW TABLES ;
+------------------+
|Tables_in_refaat  |
+------------------+
|tbref1            |
+------------------+
1 row in set (0.00 sec)

mysql>
```

- وبالنظر للشكل السابق تجد أن أعلى الجدول مكتوب:

```
Tables_in_refaat
```

وهى تدل على أن ما سيأتى بعدها هى جميع أسماء الجداول الموجودة بقاعدة البيانات المسماة refaat .

أما الاسم tbref1 فيدل على أن قاعدة البيانات refaat لا تحتوى إلا على جدول واحد فقط باسم tbref1.

* والعبارة التالية تدل على أن عملية عرض أسماء الجداول قد تمت بنجاح:

```
1 row in set (0.00 sec)
```

ملاحظة هامة:

وبالطبع قبل تنفيذ هذا الأمر تم تنفذ أمر فتح قاعد البيانات المسماة refaat بالأمر : use refaat ; ..

٥-٢-٣ عرض هيكل جدول بقاعدة البيانات المفتوحة:

الأمــر:

<DESCRIBE>

الصيــغة:

describe <table name>

مثال:

mysql > ***describe*** **refaat** ;

* وعند التنفيذ سيظهر على الشاشة شكل مشابه للشكل التالى:

```
mysql> DESCRIBE tbref1 ;
+---------+----------+------+------+----------+---------+
|Field    |Type      | Null | Key  | Default  | Extra   |
+---------+----------+------+------+----------+---------+
|name     |char(15)  | YES  |      | NULL     |         |
|address  |char(40)  | YES  |      | NULL     |         |
|salary   |float(4,2)| YES  |      | NULL     |         |
+---------+----------+------+------+----------+---------+
3 rows in set (0.05 sec)

mysql>
```

*** لاحظ:**

ظهور جدول من ٦ أعمدة يبين هيكل الجدول الحالى:

١- العمود الأول " Field ":
وهذا العمود خاص بعرض أسماء الحقول فى الجدول.

٢- العمود الثانى " Type ":
وهو خاص بعرض أنواع الحقول.

٣- العمود الثالث " Null ":
وهو يبين هل يقبل الحقل القيم الفارغة أم لا.. وفى حالة وجود " yes " داخله فهى تدل على أن هذا الحقل يقبل القيم الفارغة.

٤- العمود الرابع " Key ":
يبين هل تم تعيين الحقل كحقل مفتاح أم لا.

٥- العمود الخامس " Default ":
يبين القيم الافتراضية التى يتم تعيينها للحقل تلقائيا كلما تم إضافة سجل جديد .. وكلمة Null تعنى أن هذا الحقل لم تضاف له قيمة افتراضية.

٦- العمود السادس " Extra " :
وتبين هل يوجد وظائف إضافية للحقل أم لا.

* والعبارة التالية تدل على أن عملية عرض هيكل الجدول قد تمت بنجاح:

```
3 rows in set (0.05 sec)
```

٥-٣ مثال تطبيقى لإنشاء قاعدة بيانات و إنشاء بعض الجداول بها :

*** مثـــــال :**

أنشئ قاعدة بيانات باسم Library تحتوى علـــى ثـــلاث جـــداول مواصــــفاتها كالتالى:

book (id , name , date)
author (id , F_name , L_name)
types (id , type)

*** الحـــــل :**

- أولا :

تشغيل محرك MySQL يتم الدخول لنافـــذة الـــــ DOS والتحـــول للـــدليل bin داخل الدليل MySQL ومن ثم تنفيذ الأمر mysql:

```
C:\mysql\bin>mysql
Welcome to the MySQL monitor.  Commands with end  ; or  \g.
Your MySQL connection id is 4 to server version: 3.23.33

Type 'help; ' or  '\h' for help. Type '\c' to clear the buffer

mysql>
```

- ثانيا:

البدء فى إنشاء قاعدة البيانات Library عن طريق كتابة الأمر التالى:

Create database library;

وعند التنفيذ يظهر شكل مشابه للشكل التالى:

```
mysql> CREATE DATABASE library;
Query OK, 1 row affected (0.06 sec)

mysql>
```

- ثالثا:

فتح قاعدة البيانات لبدء استخدامها والتعامل ومعها وإنشاء جداولها.. وذلك عن طريق الأمر:

USE library ;

وعند التنفيذ يظهر الشكل التالى:

```
mysql> USE library ;
Database changed
mysql>
```

- رابعاً:

*** البدء فى إنشاء الجداول واحد يلى الأخر:**

أ) - ولنبدأ بالجدول الأول

book (id , name , date)

وهو باسم Book ، ويحتوى على ٣ حقول.

الحقل الأول :

يسمى id وهو يدل على الرقم المسلسل والفريد (غير المكرر) لكل كتاب يتم إضافته لهذا الجدول، لهذا يجب استخدام الأمر auto_increment (ترقيم تلقائى) حتى يتم إضافة رقم تلقائى لكل كتاب جديد يتم إضافته للجدول دون تدخل من المستخدم، وهذا الرقم سيكون فى تزايد مع تكرار عملية الإدخال أى (1,2,3,4,5.....)، وأيضا لجعله رقم فريد فسيتم إعطاء هذا الحقل الصفة المتاح الأساسى Primary Key، وبالطبع سيكون من النوع الرقمى.

أما الحقل الثانى:

فيسمى name وهو خاص باسم الكتاب.. وسيكون من النوع الحرفى، وسيتم وضع عدد ٣٥ حرفاً كحد أقصى لطول الاسم.. وأيضا سيتم جعل هذا الحقل لا يقبل تركه فارغا (يجب وضع قيمة به)عن طريق الأمر Not Null .

والحقل الثالث والأخير:

date فهو خاص بتاريخ إصدار الكتاب .. وسيكون من النوع التاريخى، حيث أنه سيحوى على تاريخ، ولكن هذا الحقل لن نعطه الصفة Not Null حيث انه يمكن ان يتم تسجيل أحد الكتب دون معرفة تاريخ إصدارها.

ولنبدأ التنفيذ:

- الأمر:

CREATE TABLE book (
id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
name CHAR(35) NOT NULL,
date DATE);

وعند التنفيذ يظهر شكل مشابه للشكل التالى:

```
mysql> CREATE TABLE book (
    ->id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
    ->name CHAR(35) NOT NULL,
    ->date DATE );
Query OK, 0 rows affected (0.05 sec)

mysql>
```

وبظهور العبارة:

```
Query OK, 0 rows affected (0.05 sec)
```

دل ذلك على أن الجدول قد تم إنشاءه بنجاح.

وللتأكد من ذلك أكثر.. يمكن استخدام الأمر:

DESCRIBE book ;

وعند التنفيذ يظهر شكل مشابه للشكل التالى:

```
mysql> DESCRIBE book ;
+--------+---------+-------+-----+---------+----------------+
| Field  | Type    | Null  | Key | Default | Extra          |
+--------+---------+-------+-----+---------+----------------+
| id     | int(11) |       | PRI | NULL    | auto_increment |
| name   | char(35)|       |     |         |                |
| date   | date    | YES   |     | NULL    |                |
+--------+---------+-------+-----+---------+----------------+
3 rows in set (0.00 sec)

mysql>
```

وكما هو ملاحظ أن تفاصيل وصفات جميع الحقول واضحة فى الجدول السابق..

ومن هنا يمكن التأكد من مواصفات الحقول التى تم إدخالها أثناء إنشاء الجدول.

وبعد إنشاء الجدول الأول يتم البدء فى إنشاء الجدول الثانى:

ب) إنشاء الجدول الثانى

author (id , F_name , L_name)

وهو يحتوى على ٣ حقول.

الحقل الأول:

يسمى *id*، وهو حقل عن طريقه يتم إعطاء قيمة مسلسلة فريدة لكل مؤلف.. ولجعل هذه القيمة تلقائية الإدخال يتم تخصيص الصفة *auto_increment* لها، وبهذا يكون هذا الحقل حقل مفتاح من النوع الرقمى الذى لا يقبل القيم الفارغة والتى لا تتكرر فيه القيم أكثر من مرة.

أما الحقل الثانى:

ويسمى *F_name* وهو خاص بالاسم الأول للمؤلف.. لذلك سيكون نوعه من النوع الحرفى الذى يقبل على الأكثر ٢٥حرفا، ولكنه لا يقبل القيم الفارغة.

والحقل الثالث:

المسمى *L_name* فهو خاص باسم المؤلف الأخير أى اللقب، وأيضا سيكون نوعه من النوع الحرفى الذى يقبل على الأكثر ٢٥حرفا، ولكنه لا يقبل القيم الفارغة.

ولنبدأ التنفيذ:

الأمر:

```
CREATE TABLE author (
    id INT NOT NULL PRIMARY KEY  AUTO_INCREMENT,
    F_name CHAR(25) NOT NULL,
    L_name CHAR(25) NOT NULL );
```

وعند التنفيذ يظهر شكل مشابه للشكل التالى:

```
mysql> CREATE TABLE author (
     ->id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
     ->F_name CHAR(25) NOT NULL,
     ->L_name CHAR(25) NOT NULL);
Query OK, 0 rows affected (0.06 sec)

mysql>
```

وكما قلنا سابقاً فالعبارة:

```
Query OK, 0 rows affected (0.06 sec)
```

تدل على أن الجدول قد تم إنشائه بنجاح.

وبالطبع يمكن التأكد من مواصفات الحقول بالأمر:

DESCRIBE author ;

كما حدث مع الجدول السابق.

وبعد إنشاء الجدولين السابقين *book, author* داخل قاعدة البيانات *library*

يمكن كتابة الأمر:

SHOW TABLES ;

لرؤية أسماء الجداول التى تم إضافتها فى الخطوات السابقة.

وعند تنفيذ الأمر يظهر شكل مشابه للشكل التالى:

```
mysql> SHOW TABLES ;
+-------------------+
| Tables_in_library |
+-------------------+
| auther            |
| book              |
+-------------------+
2 rows in set (0.00 sec)

mysql>
```

وكما هو ملاحظ فهو يحوى أسماء الجدولين الذى تم إنشائهم سابقا.

MR

بقى الآن الجدول الثالث:

جـ) إنشاء الجدول الثالث

types (book_id , type)

وهو باسم types ، ويحتوى على حقلين.

الحقل الأول:
يسمى book_id وهو عبارة عن رقم الكتاب الذى أضيف فى الجدول book ، وذلك لتحديد نوع هذا الكتاب، وبالطبع سيكون من النوع الرقمى الذى لا يقبل القيم الفارغة.

الحقل الثانى:
type وهو حقل نوع الكتاب، وسيتم جعله من الحرفى الذى يستوعب مالا يزيد عن ١٥ حرفاً، والذى لا يقبل القيم الفارغة أيضاً.

ولنبدأ التنفيذ :

الأمر :

```
CREATE TABLE types (
book_id INT NOT NULL,
type CHAR(15) NOT NULL );
```

وعند التنفيذ يظهر شكل مشابه للشكل التالى:

```
mysql> CREATE TABLE types (
    -> book_id INT NOT NULL ,
    -> type CHAR (15) NOT NULL );
Query OK, 0 rows affected (0.00 sec)

mysql>
```

وبظهور العبارة:

```
Query OK, 0 rows affected (0.00 sec)
```

دل ذلك على أن عملية إنشاء الجدول قد تمت بنجاح.

وبهذا تكون عملية إنشاء الجداول الثلاثة قد انتهت .. ولرؤية جميع أسماء الجداول يتم تنفيذ الأمر التالى :

SHOW TABLES ;

وعند ذلك يظهر شكل مشابه للشكل التالى:

```
mysql> SHOW TABLES ;
+-------------------+
| Tables_in_library |
+-------------------+
| auther            |
| book              |
| types             |
+-------------------+
3 rows in set (0.00 sec)

mysql>
```


MR

الفصل السادس

التعامل مع هيكل الجدول

يتناول هذا الفصل المحتويات الآتية :

- تعديل الجداول بعد إنشائها
- التعبيرات التى يمكن استخدامها مع الجملة ALTER TABLE
- أمثلة

٦ التعامل مع هيكل الجدول

٦-١ تعديل الجداول بعد إنشائها:

الأمــر:

ALTER TABLE

الصيــغة:

ALTER TABLE <table name> < alteration > [,alteration…]

الصيغة بالعربية للتوضيح:

ALTER TABLE <اسم الجدول> < نوع التعديل >
[لإضافة أكثر من تعديل فى نفس الوقت....]

مثال١:

mysql> ***ALTER TABLE*** tbref1
ADD age char(3) ***AFTER*** address;

وعند التنفيذ سيظهر على الشاشة شكل مشابه للشكل التالى:

```
mysql> ALTER TABLE tbref1 ADD age char(3) AFTER address;
Query OK, 0 rows affected (0.00 sec)
Reccrds: 0  Duplicates: 0  Warnings: 0

mysql>
```

وبظهور العبارتين:

```
Query OK, 0 rows affected (0.00 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

دل ذلك على أن العملية قد تمت بنجاح.

والأمر السابق خاص بإضافة حقل باسم **age** يستوعب ٣ أحرف على الأكثر ويكون بعد الحقل المسمى **address**.

مثال٢:

mysql> ***ALTER TABLE*** tbref1
modify age char(2) not null;

وعند التنفيذ سيظهر على الشاشة شكل مشابه للشكل التالى:

```
mysql> alter table tbref1 modify age char(2) not null ;
Query OK, 0 rows affected (0.11 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>
```

والأمر السابق خاص بتعديل الحقل **age** بحيث يستوعب حرفان على الأكثر بدلاً من ٣ أحرف، ولا يقبل القيم الفارغة.

٦-٢ والجدول الآتى يوضح التعبيرات التى يمكن استخدامها مع الجملة

ALTER TABLE

الأمر	الوظيفة
ADD Column column_description [**first** | **after** column]	١- يضيف حقل (عمود) جديد فى المكان المحدد أما فى حالة عدم تحديد مكان فسيضاف الحقل فى نهاية الجدول.
ADD (column & description , column & description …)	٢- يضيف حقل جديد واحد أو أكثر فى نهاية الجدول.

MR

ADD INDEX (column , …)	٣- لإضافة فهرس للجدول على الحقل أو الحقول المحددة.
ADD PRIMARY KEY (column,…)	٤- يجعل الحقل أو الحقول المحددة مفتاح رئيسي للجدول، بشرط أن تكون الخاصية Not Null محددة لتلك الحقول.
ADD UNIQUE (column , …)	٥- لإضافة فهرساً فريداً للجدول على الحقل أو الحقول المحددة، ويمكن تنفيذ ذلك على الحقول التي تقبل القيم الفارغة.
ALTER column SET DEFAULT value | DROP DEFAULT	٦- يضيف أو يزيل قيمة افتراضية لحقل معين.
CHANGE column new_column_description	٧- لتغيير مواصفات حقل أو لتغيير اسم حقل معين.. حيث أن new_column_description سيكتب فيه اسم الحقل.
MODIFY column column_des.	٨- مشابه لـ Change ولكنه يستخدم لتغيير أنواع الحقول وليس أسمائها.
DROP column_name	٩- لحذف أحد حقول الجدول.

DROP PRIMARY KEY	١٠- حذف صفة الفهـرس الرئيسى مـن الحقـل وليس الحقل نفسه.
DROP INDEX index_name	١١- لحذف صفة الفهـرس من الحقل المحدد.
RENAME new_table_name	١٢- لتغيير اسم الجدول.

٣-٦ أمثلة على أوامر الجدول السابق:

١- لإضافة حقل:

ALTER TABLE tbref1
ADD Work_add char(35) ;

ملاحظة:

قبل تنفيذ الأمثلة قم بفتح قاعدة البيانات *refaat* التى قمنا بإنشـائها سـابقا، ويتم الفتح عن طريق الأمر ; USE *refaat*

وعند التنفيذ يظهر الشكل التالى:

```
mysql> ALTER TABLE tbref1
       ADD Work_add char(35) NOT NULL;
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>
```

وللتأكد يتم تنفيذ الأمر:

DESCRIBE tbref1 ;

وعند التنفيذ يظهر الأتى:

```
mysql> DESCRIBE tbref1 ;
+--------+----------+------+------+----------+---------+
|Field   |Type      | Null | Key  | Default  | Extra   |
+--------+----------+------+------+----------+---------+
|name    |char(15)  | YES  |      | NULL     |         |
|address |char(40)  | YES  |      | NULL     |         |
|salary  |float(4,2)| YES  |      | NULL     |         |
|Work_add|char(35)  |      |      |          |         |
+--------+----------+------+------+----------+---------+
4 rows in set (0.00 sec)

mysql>
```

لاحظ انه تم إضافة الحقل فى نهاية الجدول.. وذلك لعدم استخدام أى من After أو First أما عند فى حالة جعل الأمر السابق كالتالى:

```
mysql> ALTER TABLE tbref1
       ADD Work_add char(35) NOT NULL
       AFTER address;
```

سيتم إضافة الحقل بعد الحقل address وقبل الحقل salary ..

٢- إضافة أكثر من حقل فى أمر واحد:

```
ALTER TABLE tbref1
        ADD ( phone char(6) NOT NULL ,
              mobile char(10) ) ;
```

- وعند التنفيذ يكون:

```
mysql> ALTER TABLE tbref1
    -> ADD (phone char(6)NOT NULL ,
    -> mobile char(10));
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>
```

وللتأكد يتم تنفيذ الأمر:

DESCRIBE tbref1 ;

```
mysql> DESCRIBE tbref1 ;
+--------+----------+------+-----+---------+---------+
|Field   |Type      | Null | Key | Default | Extra   |
+--------+----------+------+-----+---------+---------+
|name    |char(15)  | YES  |     | NULL    |         |
|address |char(40)  | YES  |     | NULL    |         |
|salary  |float(4,2)| YES  |     | NULL    |         |
|Work_add|char(35)  |      |     |         |         |
|phone   |char(6)   |      |     |         |         |
|mobile  |char(10)  | YES  |     | NULL    |         |
+--------+----------+------+-----+---------+---------+
6 rows in set (0.00 sec)

mysql>
```

٣- إضافة فهرس للجدول:

لإضافة فهرس للجدول على الحقل المسمى name يتم اســـتخدام الأمـــر التالى:

ALTER TABLE tbref1
ADD INDEX (name);

- لاحظ أن اسم الحقل يقع بين قوسين

- وعند التنفيذ يكون:

```
mysql> ALTER TABLE tbref1
    -> ADD INDEX (name);
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>
```

وللتأكد يتم تنفيذ الأمر:

DESCRIBE tbref1 ;

```
mysql> DESCRIBE tbref1 ;
+---------+----------+------+------+---------+---------+
|Field    |Type      | Null | Key  | Default | Extra   |
+---------+----------+------+------+---------+---------+
|name     |char(15)  | YES  | MUL  | NULL    |         |
|address  |char(40)  | YES  |      | NULL    |         |
|salary   |float(4,2)| YES  |      | NULL    |         |
|Work_add |char(35)  |      |      |         |         |
|phone    |char(6)   |      |      |         |         |
|mobile   |char(10)  | YES  |      | NULL    |         |
+---------+----------+------+------+---------+---------+
6 rows in set (0.00 sec)

mysql>
```

MR

لاحظ وجود كلمة MUL داخل العمود " Key " فى الحقل " name " للدلالة على أن الجدول مفهرس بهذا الحقل.

٤- إضافة مفتاح رئيسى للجدول :

لإضافة مفتاح رئيسى للجدول على الحقل المسمى phone يتم استخدام الأمر التالى :
ولكن يجب ملاحظة أن الحقل المراد جعله مفتاح أساسى يجب أن تكون الصفة NOT NULL مضافة له، ولن يتمكن MySQL من جعل الحقل مفتاح أساسى إذا كان الحقل المراد التنفيذ عليه يقبل القيم الفارغة.. وهذا يفسر أننا عند إنشاء الحقل phone أعطينا الصفة NOT NULL له.

ALTER TABLE tbref1
ADD PRIMARY KEY (phone);

- لاحظ أن اسم الحقل يقع بين قوسين.

- وعند التنفيذ يكون:

```
mysql> ALTER TABLE tbref1
    -> ADD PRIMARY KEY (phone);
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>
```


MR

وللتأكد يتم تنفيذ الأمر:

DESCRIBE tbref1 ;

```
mysql> DESCRIBE tbref1 ;
+--------+----------+------+-----+---------+---------+
|Field   |Type      | Null | Key | Default | Extra   |
+--------+----------+------+-----+---------+---------+
|name    |char(15)  | YES  | MUL | NULL    |         |
|address |char(40)  | YES  |     | NULL    |         |
|salary  |float(4,2)| YES  |     | NULL    |         |
|Work_add|char(35)  |      |     |         |         |
|phone   |char(6)   |      | PRI |         |         |
|mobile  |char(10)  | YES  |     | NULL    |         |
+--------+----------+------+-----+---------+---------+
6 rows in set (0.00 sec)

mysql>
```

لاحظ وجود كلمة PRI داخل العمود " Key " فى الحقل " phone " للدلالـــة على أن هذا الحقل مفتاح أساسى بالجدول.

٥- إضافة فهرس فريد للجدول:

لإضافة فهرس فريد للجدول على الحقل المســمى mobile يــتم اســتخدام الأمر التالى:

ALTER TABLE tbref1
ADD UNIQUE (mobile);

- لاحظ أن اسم الحقل يقع بين قوسين

- وعند التنفيذ يكون:

```
mysql> ALTER TABLE tbref1
    -> ADD UNIQUE (mobile);
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>
```

وللتأكد يتم تنفيذ الأمر:

DESCRIBE tbref1 ;

```
mysql> DESCRIBE tbref1 ;
+--------+----------+------+------+---------+---------+
|Field   |Type      | Null | Key  | Default | Extra   |
+--------+----------+------+------+---------+---------+
|name    |char(15)  | YES  | MUL  | NULL    |         |
|address |char(40)  | YES  |      | NULL    |         |
|salary  |float(4,2)| YES  |      | NULL    |         |
|Work_add|char(35)  |      |      |         |         |
|phone   |char(6)   |      | PRI  |         |         |
|mobile  |char(10)  | YES  | UNI  | NULL    |         |
+--------+----------+------+------+---------+---------+
6 rows in set (0.00 sec)

mysql>
```

لاحظ وجود كلمة **UNI** داخل العمود " Key " فى الحقل " mobile " للدلالة على أن هذا الحقل فهرس فريد بالجدول.

٦- إضافة أو إزالة قيمة افتراضية لحقل معين:

*** القيمة الافتراضية هى:**

القيمة التى تظهر فى الحقل فى حالة عدم إدخال أى بيانات فى هذا الحقل.

ولإضافة قيمة افتراضية لأحد حقول الجدول وليكن على الحقل المسمى address يتم استخدام الأمر التالى:

ALTER TABLE tbref1
ALTER address **SET DEFAULT** “Damitta”;

- وعند التنفيذ يكون:

```
mysql> ALTER TABLE tbref1
    -> ALTER address SET DEFAULT "Damitta";
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>
```

لاحظ أننا وضعنا القيمة الافتراضية بين علامتى تنصيص لأنها نصية.

وللتأكد يتم تنفيذ الأمر:

DESCRIBE tbref1 ;

```
mysql> DESCRIBE tbref1 ;
+--------+----------+------+------+----------+---------+
|Field   |Type      | Null | Key  | Default  | Extra   |
+--------+----------+------+------+----------+---------+
|name    |char(15)  | YES  | MUL  | NULL     |         |
|address |char(40)  | YES  |      | Damitta  |         |
|salary  |float(4,2)| YES  |      | NULL     |         |
|Work_add|char(35)  |      |      |          |         |
|phone   |char(6)   |      | PRI  |          |         |
|mobile  |char(10)  | YES  | UNI  | NULL     |         |
+--------+----------+------+------+----------+---------+
6 rows in set (0.06 sec)

mysql>
```

لاحظ أن كلمة Damitta ظهرت فى العمود المسمى Default فى الحقل address مما يدل على أنه فى حالة عدم إضافة أى قيمة للحقل address سيتم وضع كلمة Damitta.

* ولإلغاء قيمة افتراضية لأحد حقول الجدول وليكن على الحقل address الذى سبق ووضعنا له القيمة الافتراضية Damitta يتم استخدام الأمر التالى :

```
ALTER TABLE tbref1
            ALTER address DROP DEFAULT;
```


MR

- وعند التنفيذ يكون:

```
mysql> ALTER TABLE tbref1
    -> ALTER address DROP DEFAULT ;
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>
```

- وللتأكد يتم تنفيذ الأمر:

DESCRIBE tbref1 ;

```
mysql> DESCRIBE tbref1 ;
+--------+----------+------+------+----------+---------+
|Field   |Type      | Null | Key  | Default  | Extra   |
+--------+----------+------+------+----------+---------+
|name    |char(15)  | YES  | MUL  | NULL     |         |
|address |char(40)  | YES  |      | NULL     |         |
|salary  |float(4,2)| YES  |      | NULL     |         |
|Work_add|char(35)  |      |      |          |         |
|phone   |char(6)   |      | PRI  |          |         |
|mobile  |char(10)  | YES  | UNI  | NULL     |         |
+--------+----------+------+------+----------+---------+
6 rows in set (0.06 sec)

mysql>
```

لاحظ أن كلمة Damitta اختفت من العمود المسمى Default في الحقل address وظهرت بدلا منها الكلمة NULL مما يدل على أن القيمة الافتراضية قد تم إلغائها من الحقل address.

٧- لتغيير مواصفات حقل أو لتغيير اسم حقل معين:

فإذا أرنا تغيير اسم الحقل salary إلى الاسم p_salary فسيتم استخدام الأمر التالى:

```
ALTER TABLE tbref1
        CHANGE salary  p_salary float(4,2);
```

- وعند التنفيذ يكون:

```
mysql> ALTER TABLE tbref1
    -> CHANGE salary p_salary float(4,2) ;
Query OK, 0 rows affected (0.11 sec)

mysql>
```

وللتأكد يتم تنفيذ الأمر:

```
DESCRIBE tbref1 ;
```

```
mysql> DESCRIBE tbref1 ;
+---------+----------+------+------+----------+---------+
|Field    |Type      | Null | Key  | Default  | Extra   |
+---------+----------+------+------+----------+---------+
|name     |char(15)  | YES  | MUL  | NULL     |         |
|address  |char(40)  | YES  |      | NULL     |         |
|p_salary |float(4,2)| YES  |      | NULL     |         |
|Work_add |char(35)  |      |      |          |         |
|phone    |char(6)   |      | PRI  |          |         |
|mobile   |char(10)  | YES  | UNI  | NULL     |         |
+---------+----------+------+------+----------+---------+
6 rows in set (0.06 sec)

mysql>
```


MR

- لاحظ انه لتغير الاسم تم كتابة مواصفات الحقل كاملة.

وعند الرغبة فى تغيير المواصفات الخاصة بالحقل فقط ، وليكن p_salary وتحويله من float(4,2) إلى float(5,3) يتم تنفيذ الأمر التالى :

ALTER TABLE tbref1
CHANGE p_salary p_salary float(5,3);

- وعند التنفيذ يكون:

```
mysql> ALTER TABLE tbref1
    -> CHANGE p_salary p_salary float(5,3) ;
Query OK, 0 rows affected (0.05 sec)

mysql>
```

وللتأكد يتم تنفيذ الأمر:

DESCRIBE tbref1 ;

```
mysql> DESCRIBE tbref1 ;
+--------+----------+-----+-----+---------+---------+
|Field   |Type      | Null | Key | Default | Extra   |
+--------+----------+-----+-----+---------+---------+
|name    |char(15)  | YES | MUL | NULL    |         |
|address |char(40)  | YES |     | NULL    |         |
|p_salary|float(5,3)| YES |     | NULL    |         |
|Work_add|char(35)  |     |     |         |         |
|phone   |char(6)   |     | PRI |         |         |
|mobile  |char(10)  | YES | UNI | NULL    |         |
+--------+----------+-----+-----+---------+---------+
6 rows in set (0.06 sec)

mysql>
```

- لاحظ أيضاً انه لتغير مواصفات الحقل يتم كتابة مواصفات الحقل كاملة.

٨- لتغيير مواصفات حقل:

مع الأمر السابق CHANGE كان بالإمكان تغيير اسم الحقل بالإضافة لمواصفاته، أما مع هذا الأمر MODIFY فيمكن تغيير مواصفات الحقل فقط دون إمكانية تغيير اسمه.

ولتغيير مواصفات الحقل name وجعله يستوعب ٢٥ حرفاً بدلاً من ١٥ حرفا يتم استخدام الأمر التالى:

```
ALTER TABLE tbref1
        MODIFY name char(25);
```

- وعند التنفيذ يكون:

```
mysql> ALTER TABLE tbref1
    -> MODIFY name char(25) ;
Query OK, 0 rows affected (0.00 sec)

mysql>
```

وللتأكد يتم تنفيذ الأمر:

```
DESCRIBE tbref1 ;
```

```
mysql> DESCRIBE tbref1 ;
+--------+----------+------+------+----------+---------+
|Field   |Type      | Null | Key  | Default  | Extra   |
+--------+----------+------+------+----------+---------+
|name    |char(25)  | YES  | MUL  | NULL     |         |
|address |char(40)  | YES  |      | NULL     |         |
|p_salary|float(5,3)| YES  |      | NULL     |         |
|Work_add|char(35)  |      |      |          |         |
|phone   |char(6)   |      | PRI  |          |         |
|mobile  |char(10)  | YES  | UNI  | NULL     |         |
+--------+----------+------+------+----------+---------+
6 rows in set (0.06 sec)

mysql>
```

٩- لحذف أحد حقول الجدول:

والمثال التالى يوضح كيفية استخدام الأمر:

*** مثال:**

قم بفتح قاعدة البيانات refaat وإضافة الحقل الرقمى الصحيح tax والـذى يستوعب ٥ أرقام فى الجدول tbref1 بعد الحقل المسمى **address** وتأكد من ذلك ، ومن ثم قم بإلغاء هذا الحقل..

*** الحل:**

أ) لفتح قاعدة البيانات يستخدم الأمر التالى:

USE refaat ;

وعند التنفيذ يظهر:

```
mysql> USE refaat ;
Database changed
mysql>
```

ب) إضافة الحقل tax:

بما أن الحقل رقمى صحيح إذن فانه سيأخذ النوع integer ، وبما انه بعد الحقل **address** فسنستخدم الأمر AFTER مع الأمر ADD ؛

ولتنفيذ ذلك يتم كتابة الأمر التالى :

```
ALTER TABLE tbref1
        ADD tax int(5) AFTER address;
```

وعند التنفيذ يظهر :

```
mysql> ALTER TABLE tbref1
       ADD tax int(5) AFTER address;
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>
```

ج) وللتأكد يتم تنفيذ الأمر:

DESCRIBE tbref1 ;

```
mysql> DESCRIBE tbref1 ;
+---------+----------+------+-----+---------+---------+
|Field    |Type      | Null | Key | Default | Extra   |
+---------+----------+------+-----+---------+---------+
|name     |char(25)  | YES  | MUL | NULL    |         |
|address  |char(40)  | YES  |     | NULL    |         |
|tax      |int(5)    | YES  |     | NULL    |         |
|p_salary |float(5,3)| YES  |     | NULL    |         |
|Work_add |char(35)  |      |     |         |         |
|phone    |char(6)   |      | PRI |         |         |
|mobile   |char(10)  | YES  | UNI | NULL    |         |
+---------+----------+------+-----+---------+---------+
7 rows in set (0.00 sec)

mysql>
```

ء) حذف الحقل المسمى tax:

ولحذف الحقل يتم تنفيذ الأمر التالى:

```
ALTER TABLE tbref1
            DROP  tax ;
```

وعند التنفيذ يظهر:

```
mysql> ALTER TABLE tbref1 DROP tax ;
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>
```

MR

وبهذا يكون قد تم حذف الحقل المسمى tax من الجــدول tbref1 الموجــود بقاعدة البيانات refaat.

١٠- حذف الفهرس الرئيسى من الجدول:

ولحذف الفهرس الرئيسى من على الحقل phone يتم تنفيذ الأمر التالى:

```
ALTER TABLE tbref1
            DROP PRIMARY KEY ;
```

وعند التنفيذ يظهر:

```
mysql> ALTER TABLE tbref1
    -> DROP PRIMARY KEY;
Query OK, 0 rows affected (0.00 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>
```

وللتأكد يتم تنفيذ الأمر:

```
DESCRIBE tbref1 ;
```

```
mysql> DESCRIBE tbref1 ;
+---------+----------+------+-----+---------+--------+
|Field    |Type      | Null | Key | Default | Extra  |
+---------+----------+------+-----+---------+--------+
|name     |char(25)  | YES  | MUL | NULL    |        |
|address  |char(40)  | YES  |     | NULL    |        |
|p_salary|float(5,3)| YES  |     | NULL    |        |
|Work_add|char(35)  |      |     |         |        |
|phone    |char(6)   |      |     |         |        |
|mobile   |char(10)  | YES  | UNI | NULL    |        |
+---------+----------+------+-----+---------+--------+
7 rows in set (0.00 sec)

mysql>
```

بهذا سيتم إلغاء الفهرس الرئيسى من على الحقول التى تحتوى عليه.

MR

١١- لحذف الفهرس من الحقل المحدد:

والمقصود بالفهرس هنا هو الفهرس الذى تم إضافته على أحد الحقول عـن طريق الأمر:

ADD INDEX (column_name)

وقد أضفناه كما جاء فى الصفحات السابقة على الحقل name.

ولحذفه يتم تنفيذ الأمر التالى:

ALTER TABLE tbref1
DROP INDEX name ;

وعند التنفيذ يظهر:

```
mysql> ALTER TABLE tbref1 DROP INDEX name ;
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>
```

بهذا يكون الفهرس الموجود على الحقل name قد حذف.

والشكل التالى يوضح تأثير أوامر حذف المفتاح الرئيسى والفهرس من الجدول:

```
mysql> DESCRIBE tbref1 ;
+---------+----------+------+------+----------+----------+
|Field    |Type      | Null | Key  | Default  | Extra    |
+---------+----------+------+------+----------+----------+
|name     |char(25)  | YES  |      | NULL     |          |
|address  |char(40)  | YES  |      | NULL     |          |
|p_salary|float(5,3)| YES  |      | NULL     |          |
|Work_add|char(35)  |      |      |          |          |
|phone    |char(6)   |      |      |          |          |
|mobile   |char(10)  | YES  | UNI  | NULL     |          |
+---------+----------+------+------+----------+----------+
6 rows in set (0.00 sec)

mysql>
```

- وإذا أردنا أيضا حذف الفهرس الفريد الموجود فى الحقـل المسـمى mobile فباستخدام الأمر التالى يتم حذف هذا الفهرس الفريد أيضا:

ALTER TABLE tbref DROP INDEX mobile ;

وعند التنفيذ يظهر:

```
mysql> ALTER TABLE tbref1 DROP INDEX mobile;
Query OK, 0 rows affected (0.00 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>
```


MR

١٢- لتغيير اسم الجدول:

ربما يحتاج المستخدم لتغيير اسم الجدول من اسم إلى أخر ... ومن خـــلال هذا الأمر يمكن تغيير أسماء الجداول الموجود بقاعدة البيانات..

ولتغيير اسم الجدول tbref1 إلى الاسم tbref (وذلك بحذف الـــرقم 1 الـــذى يحتوى عليه اسم الجدول) يتم استخدام الأمر التالى:

ALTER TABLE tbref1 RENAME tbref ;

وعند التنفيذ يظهر:

```
mysql> ALTER TABLE tbref1 RENAME tbref ;
Query OK, 0 rows affected (0.06 sec)

mysql>
```

وللتأكد من تغيير الاسم يتم تنفيذ الأمر :

SHOW TABLES ;

وعند التنفيذ يظهر:

```
mysql> SHOW TABLES ;
+------------------+
|Tables_in_refaat  |
+------------------+
|tbref             |
+------------------+
1 row in set (0.00 sec)

mysql>
```

الفصل السابع

التعامل مع البيانات
(إدخال - عرض - تعديل - حذف)

يتناول هذا الفصل المحتويات الآتية :

- الإدخال للبيانات
- عرض البيانات
- التعديل فى البيانات
- عمليات الحذف المختلفة

MR

٧-١ الإدخال للبيانات

إضافة بيانات جديدة للحقول:

الأمر:

< INSERT INTO >

الصيغة:

```
INSERT [INTO] table [(column1,column2,...)]
        VALUES (value1, value2, ... )
```

الصيغة بالعربية للتوضيح:

INSERT [INTO] اسم الجدول [(اسماء الحقول المراد الإضافة بها)]
VALUES [(البيانات المراد إضافتها بنفس ترتيب الحقول)]

لاحظ انه يجب كتابة جميع الحقول المراد الإضافة بها مفصول بينهم بفاصلة.. وكتابة البيانات المراد إضافتها بحيث يكون ترتيبها نفس ترتيب الحقول ويجب مراعاة وضع البيانات النصية بين علامتى تنصيص.

*** مثال:**

أضف البيانات التالية داخل الجدول tbref :

name = Mohamed
address = Damitta
p_salary = 111.5
Work_add = New Damitta
Phone = 112233
Mobile = 0199999999

*** الحـــــل:**

أ) يتم استخدام الأمر التالى:

```
INSERT INTO tbref (name, address, p_salary, work_add, phone, mobile)
VALUES ("mohamed","Damitta",111.5, "New Damitta",
"112233","0199999999");
```

وعند التنفيذ يظهر :

```
mysql> INSERT INTO tbref (name, address, p_salary,
work_add, phone, mobile) VALUES
("mohamed","Damitta",111.5,"New Damitta",
"112233","0199999999");
Query OK, 1 rows affected (0.05 sec)

mysql>
```

وبظهور الرسالة التالية:

```
Query OK, 1 rows affected (0.05 sec)
```

دل ذلك على أن عملية الإضافة تمت بنجاح.

MR

ب) طريقة أخرى للإضافة:

يمكن إضافة البيانات دون كتابة أسماء الحقول وذلك فى حالة أن البيانات المدخلة ستضاف للحقول حسب ترتيبها.. أى أن القيمة المذكورة أولا ستضاف فى الحقل الأول والثانية فى الحقل الثانى وهكذا...

```
mysql> INSERT INTO tbref VALUES
("mohamed","Damitta",111.5,"New Damitta",
"112233","0199999999");
Query OK, 1 rows affected (0.05 sec)

mysql>
```

جـ) طريقة أخرى للإضافة:

يمكن استخدام الطريقة التالية للإدخال بدلاً من الطريقة السابقة :

```
INSERT INTO tbref
SET    name          ="Mohamed",
       Address       ="Damitta",
       p_salary =111.5,
       work_add ="Nea Damitta",
       phone    ="112233",
       mobile        ="0199999999" ;
```

وبالطبع يمكن استخدام أحد الطرق السابقة لإدخال البيانات بالجداول.. وذلك حسب رغبة المستخدم.

٧-٢ عرض البيانات

يستخدم الأمر ***SELECT*** للقيام بجميع عمليات العرض المختلفة للبيانات.. أى عن طريقه يتم عرض جميع بيانات جدول أو عرض مجموعة من البيانــات فقط.. أو عرض جزء من السجلات أو غير ذلك من جميع العمليات المختلفة.

٧-٢-١ عرض جميع محتويات أحد الجداول:

الأمــر:

SELECT *

الصيــغة :

SELECT <*> FROM <table name>

مثال:

لإظهار جميع بيانات الجدول tbref يتم استخدام الأمر التالى:

SELECT * FROM tbref ;

وعند التنفيذ يظهر الشكل التالى:

MR

```
mysql> SELECT * FROM tbref ;
+---------+--------+---------+------------+--------+-----------+
| name    | address| p_salary | Work_add   | phone  | mobile    |
+---------+--------+---------+------------+--------+-----------+
| mohamed | Damitta| 111.500 | New Damitta | 112233 | 0199999999 |
+---------+--------+---------+------------+--------+-----------+
1 row in set (0.06 sec)

mysql>
```

لاحظ أنه تم عرض جميع حقول الجدول.

والسطر التالى:

```
1 row in set (0.06 sec)
```

يدل على وجود سجل (صف) واحد فقط تم عرضه.

٧-٢-٢ عرض حقول معينة دون غيرها:

أى [illegible] الرغبة فى عرض حقل name بالإضافة لحقل phone يتم استخدام أيضاً الأمر SELECT ولكن مضاف له شرط.

الأمر:

SELECT ***items***

الصيغة:

SELECT <item1,item2, ... > **FROM** <table name>

والكلمة items هنا تعبر عن الحقل أو الحقول المراد عرض محتوياتها دون غيرها.

مثال :

فى إذا أردنا تنفيذ العرض لحقل name والحقل phone فقط ... يتم كتابة الأمر التالى:

SELECT name, phone **FROM** tbref ;

وعند التنفيذ يظهر الشكل التالى:

```
mysql> SELECT name, phone FROM tbref;
+--------+--------+
| name   | phone  |
+--------+--------+
| mohamed| 112233 |
+--------+--------+
1 row in set (0.00 sec)

mysql>
```

لاحظ ظهور حقلين فقط دون.. وأيضا ظهور السطر:

```
1 row in set (0.00 sec)
```

للدلالة على أن عدد السجلات (الصفوف) المعروضة تساوى ١ .. وذلك لأننا حتى الآن لم نضيف غير بيانات سجل واحد فقط.

٧-٢-٣ عرض جميع المحتويات (السجلات) بشرط معين:

الأمر:

SELECT * … … WHERE

صيــغة ::

SELECT * FROM <table name> **WHERE** <conditions>

للبدء فى تنفيذ هــذه الحالــة.. يجــب أولا إضـــافة ٤ ســجلات أخـــرى للجدول tbref الموجود فى قاعدة البيانات refaat ..

لذلك سيتم إضافة البيانات التالية للجدول:

السجل الأول:

name = Ahmed
address = Damitta
p_salary = 151.5
Work_add = New Damitta
Phone = 998877
Mobile = 0177777777

السجل الثانى :

name = Aly
address = Cairo
p_salary = 95.3
Work_add = Geza
Phone = 554466
Mobile = 0188888888

السجل الثالث :

```
name        = Khaled
address     = Aswan
p_salary    = 101.2
Work_add    = Aswan
Phone       = 448899
Mobile      = 0166666666
```

السجل الرابع :

```
name        = Zyad
address     = Tanta
p_salary    = 250.3
Work_add    = New Damitta
Phone       = 885599
Mobile      = 0195555555
```

- وسنستخدم الأمر التالى لإضافة تلك البيانات (السجلات) :

```
INSERT INTO tbref
SET
  name       = "Ahmed",
  address    = "Damitta",
  p_salary   = 151.5,
  Work_add   = "New Damitta",
  Phone      = "998877",
  Mobile     ="0177777777" ;
```

وهكذا حتى يتم إضافة جميع السجلات

مثال ١:

عند الرغبة فى عرض جميع السجلات التى يعمل أصحابها فى مدينة دمياط الجديدة .. يتم كتابة الأمر التالى:

```
SELECT *    FROM tbref
            WHERE work_add LIKE "new damitta" ;
```

وعند التنفيذ يظهر شكل مشابه للشكل التالى:

```
mysql> SELECT * FROM tbref
    ->WHERE work_add LIKE "new damitta ;
+----------+---------+----------+--------------+---------+------------+
| name     | address | p_salary | Work_add     | phone   | mobile     |
+----------+---------+----------+--------------+---------+------------+
| Ahmed    | Damitta |  151.500 | New Damitta  | 998877  | 0177777777 |
| mohamed  | Damitta |  111.500 | New Damitta  | 112233  | 0199999999 |
| Zyad     | Tanta   |  250.300 | New Damitta  | 885599  | 0195555555 |
+----------+---------+----------+--------------+---------+------------+
3 rows in set (0.00 sec)

mysql>
```

لاحظ ظهور فقط السجلات التى يعمل أصحابها فى مدينة دمياط الجديدة New Damitta ..

أما عند كتابة الأمر دون شروط فسيكون ناتج التنفيذ كالآتى :

```
mysql> SELECT * FROM tbref ;
+----------+---------+----------+--------------+---------+------------+
| name     | address | p_salary | Work_add     | phone   | mobile     |
+----------+---------+----------+--------------+---------+------------+
| Khaled   | Aswan   |  101.200 | Aswan        | 448899  | 0166666666 |
| Ahmed    | Damitta |  151.500 | New Damitta  | 998877  | 0177777777 |
| mohamed  | Damitta |  111.500 | New Damitta  | 112233  | 0199999999 |
| Aly      | Cairo   |   95.300 | Geza         | 554466  | 0188888888 |
| Zyad     | Tanta   |  250.300 | New Damitta  | 885599  | 0195555555 |
+----------+---------+----------+--------------+---------+------------+
5 rows in set (0.06 sec)

mysql>
```

لاحظ هنا ظهور كامل السجلات وجميع الحقول .. وذلك لعدم كتابة أى شرط بالأمر..

مثال ٢:

عند الرغبة فى عرض جميع السجلات التى يعمل أصحابها فى New Damitta ويسكنون فى Damitta .. يتم كتابة الأمر التالى:

```
SELECT * FROM tbref
         WHERE address LIKE "damitta"
         AND   work_add LIKE "new damitta"
```

وعند التنفيذ يظهر شكل مشابه للشكل التالى:

```
mysql> SELECT * FROM tbref
    -> WHERE address LIKE "damitta" AND work_add LIKE "new damitta ;
+---------+--------+----------+------------+--------+------------+
| name    | address| p_salary | Work_add   | phone  | mobile     |
+---------+--------+----------+------------+--------+------------+
| Ahmed   | Damitta|  151.500 | New Damitta| 998877 | 0177777777 |
| mohamed | Damitta|  111.500 | New Damitta| 112233 | 0199999999 |
+---------+--------+----------+------------+--------+------------+
2 rows in set (0.06 sec)

mysql>
```

ملاحظة :

لقد تمت استخدام الكلمة **LIKE** بدلا من علامة التساوى "=" فى المثالين السابقين ..

ذلك لأنه يفضل مع الحقول النصية استخدام **LIKE** (لاختبار تطابق قيمتين) أما مع الحقول الرقمية فتستخدم علامة التساوى " = " (لاختبار تساوى رقمين)... مع العلم أنه يمكن استخدام علامة التساوى مع الحقول النصية.

وأيضا تم استخدام **AND** للدلالة على أنه يجب أن يتوافر الشرطان فى الحقول التى سيتم عرضها ...

ويمكن أيضا استخدام **OR** للدلالة على أن الحقل سيم عرضه فى حالة توافر أحد الشرطين.

MR

مثال٣:

- لعرض جميع السجلات التى يسكن أصحابها فى Tanta أو أن اسمهم Aly.

- وللتنفيذ يتم كتابة الأمر التالى :

```
SELECT  *  FROM tbref
        WHERE address LIKE "tanta"
        OR name LIKE "aly" ;
```

وعند التنفيذ يظهر شكل مشابه للتالى:

```
MYSQL> SELECT * FROM tbref
    -> WHERE address LIKE "tanta" OR name LIKE "aly" ;
+------+--------+----------+------------+--------+------------+
| name | address| p_salary | Work_add   | phone  | mobile     |
+------+--------+----------+------------+--------+------------+
| Aly  | Cairo  | 95.300   | Geza       | 554466 | 0188888888 |
| Zyad | Tanta  | 250.300  | New Damitta| 885599 | 0195555555 |
+------+--------+----------+------------+--------+------------+
2 rows in set (0.05 sec)

mysql>
```

- لاحظ ظهور حقلين يتوفر فى كل منهما أحد الشروط.

٧-٢-٤ عرض مجموعة محددة من الحقول عند توافر شرط معين أو أكثر:

يتم التنفيذ عن طريق تحديد أسماء الحقول المراد عرضها.. وذلك بعد الأمر SELECT .

الأمر:

SELECT ***items*** **... ... WHERE**

صيغة ::

SELECT <item1,item2, ... > **FROM** <table name>
WHERE <conditions>

مثال:

لعرض الاسم والعنوان ورقم الهاتف للسجلات التى يعمل أصحابها فى New Damitta ويسكنون فى Damitta .. يتم كتابة الأمر التالى :

```
SELECT name, address, phone  FROM tbref
       WHERE address LIKE "damitta"
       AND    work_add LIKE "new damitta"
```

وعند التنفيذ يظهر شكل مشابه للشكل التالى:

MR

```
mysql> SELECT name, address, phone FROM tbref
    -> WHERE address LIKE "damitta" AND work_add LIKE "new damitta ;
+---------+--------+---------+
| name    | address| phone   |
+---------+--------+---------+
| Ahmed   | Damitta| 998877  |
| mohamed | Damitta| 112233  |
+---------+--------+---------+
2 rows in set (0.05 sec)

mysql>
```

لاحظ ظهور ٣ أعمدة (حقول) فقط مع عدم ظهور باقى الأعمدة.

ملاحظة:

يمكن التحكم فى ترتيب الأعمدة المعروضة، حيث يمكن تغيير الترتيب الافتراضى لهم عن طريق كتابة أسماء الحقول بعد الأمر SELECT حسب الترتيب المطلوب..

فمثلاً: عند الرغبة فى إظهار حقل phone أولا ثم حقل name ثم حقل address .. يتم كتابة الأمر كما يلى:

SELECT phone ,name, address **FROM** tbref
WHERE address **LIKE** "damitta"
AND work_add **LIKE** "new damitta"

وعند التنفيذ يظهر شكل مشابه للشكل التالى:

```
mysql> SELECT phone, name, address FROM tbref
    -> WHERE address LIKE "damitta" AND work_add LIKE "new damitta ;
+---------+---------+--------+
| phone   | name    | address|
+---------+---------+--------+
| 998877  | Ahmed   | Damitta|
| 112233  | mohamed | Damitta|
+---------+---------+--------+
2 rows in set (0.05 sec)

mysql>
```

لاحظ التغير الذى طرأ على أماكن الأعمدة بعد تنفيذ الأمر ، وهذا التغير بالطبع يكون عند العرض فقط أى أن طريقة العرض لن تؤثر على الترتيب الأصلى للحقول... ولكنه ترتيب أثناء العرض على الشاشة فقط.

٧-٢-٥ عرض السجلات مرتبة تبعاً لأحد الحقول:

ربما يحتاج المستخدم لعرض بيانات السجلات ولكن بشرط أن تكون مرتبة تبعاً للاسم أو المرتب أو العنوان أو أى حقل لأخر..

ولتنفيذ ذلك يتم استخدام صفة جديدة مع الأمر SELECT .. كما يلى :

الصفة (الأمر):

ORDER BY

صيغة الأمر:

SELECT items **FROM** <table name> **ORDER BY** <columns name>

*** مثال ١:**

اعرض جميع السجلات بالجدول tbref بشرط أن تكون مرتبة أبجديا تبعا لحقل الاسم name.

MR

* الحل :

يتم كتابة الأمر التالى:

SELECT * FROM tbref **ORDER BY** name ;

```
mysql> SELECT * FROM tbref ORDER BY name;
+---------+--------+---------+------------+--------+-----------+
| name    | address| p_salary | Work_add   | phone  | mobile    |
+---------+--------+---------+------------+--------+-----------+
| Ahmed   | Damitta|  151.500 | New Damitta | 998877 | 0177777777 |
| Aly     | Cairo  |   95.300 | Geza        | 554466 | 0188888888 |
| Khaled  | Aswan  |  101.200 | Aswan       | 448899 | 0166666666 |
| mohamed | Damitta|  111.500 | New Damitta | 112233 | 0199999999 |
| Zyad    | Tanta  |  250.300 | New Damitta | 885599 | 0195555555 |
+---------+--------+---------+------------+--------+-----------+
5 rows in set (0.00 sec)

mysql>
```

لاحظ الترتيب الأبجدى التصاعدى للجدول بحقل الاسم name.

* مثال ٢:

اعرض جميع السجلات بالجدول tbref بشرط أن تكون مرتبة أبجديا تبعـا لحقل المرتب p_salary .

* الحل :

يتم كتابة الأمر التالى:

SELECT * FROM tbref **ORDER BY** p_salary ;

```
mysql> SELECT * FROM tbref ORDER BY p_salary;
+---------+--------+---------+------------+--------+-----------+
| name    | address| p_salary | Work_add   | phone  | mobile    |
+---------+--------+---------+------------+--------+-----------+
| Aly     | Cairo  |  95.300 | Geza       | 554466 | 0188888888 |
| Khaled  | Aswan  | 101.200 | Aswan      | 448899 | 0166666666 |
| mohamed | Damitta| 111.500 | New Damitta | 112233 | 0199999999 |
| Ahmed   | Damitta| 151.500 | New Damitta | 998877 | 0177777777 |
| Zyad    | Tanta  | 250.300 | New Damitta | 885599 | 0195555555 |
+---------+--------+---------+------------+--------+-----------+
5 rows in set (0.06 sec)

mysql>
```

لاحظ ترتيب الجدول بحقل الاسم p_salary.

*** مثال ٣ :**

أعرض حقل الاسم فقط ورتبه أبجديا.

*** الحل:**

يتم كتابة الأمر التالى:

SELECT name **FROM** tbref **ORDER BY** name ;

وعند التنفيذ يظهر شكل مشابه للتالى:

```
mysql> SELECT name FROM tbref ORDER BY name;
+---------+
| name    |
+---------+
| Ahmed   |
| Aly     |
| Khaled  |
| mohamed |
| Zyad    |
+---------+
5 rows in set (0.00 sec)

mysql>
```

- يمكنك استعمال الكلمة الأساسية **ASC** وهو خـاص بالترتيـب التصـاعدى للأحرف أو الأرقام (وهو الافتراضى)..

مثال:

```
SELECT *
        FROM tbref
        ORDER BY name  ASC ;
```

- ويمكنك أيضاً استعمال الكلمة الأساسية **DESC** وهو خاص بالترتيب التنازلى للأحرف أو الأرقام (ترتيب معاكس)..

مثال:

```
SELECT *
FROM tbref
ORDER BY name  DESC ;
```

وعند التنفيذ يظهر التالى:

```
mysql> SELECT * FROM tbref ORDER BY name DESC;
+---------+--------+---------+------------+--------+------------+
| name    | address| p_salary| Work_add   | phone  | mobile     |
+---------+--------+---------+------------+--------+------------+
| Zyad    | Tanta  | 250.300 | New Damitta| 885599 | 0195555555 |
| mohamed | Damitta| 111.500 | New Damitta| 112233 | 0199999999 |
| Khaled  | Aswan  | 101.200 | Aswan      | 448899 | 0166666666 |
| Aly     | Cairo  |  95.300 | Geza       | 554466 | 0188888888 |
| Ahmed   | Damitta| 151.500 | New Damitta| 998877 | 0177777777 |
+---------+--------+---------+------------+--------+------------+
5 rows in set (0.00 sec)

mysql>
```

لاحظ الترتيب الأبجدى التنازلى للجدول بحقل الاسم name.
وهكذا يمكن متابعة العرض بأى ترتيب مطلوب.

٧-٢-٦ اختيار صفوف معينة لتنفيذ الأوامر عليها:

ربما يحتاج المستخدم اختيار صـــفوف معينـــة لتنفيـــذ الأوامـــر عليهـــا.. ولتنفيذ ذلك يتم استخدام صفة جديدة مع الأوامر.. كما يلى:

الصفة (الأمر):

LIMIT number1, number2

ولا يأتى هذا الأمر منفرداً بل يأتى مع أحد الأوامر الأخرى مثـــل أوامـــر العرض والتحديث والحذف ...

انظر الأمر التالى:

SELECT *items* **FROM** table name
[LIMIT number1, number2]

لاحظ انه يفصل بين الرقمين بفاصلة..

والرقم الأول number1 يدل على رقم الصف المطلوب البدء منه .
والرقم الثانى number2 يدل على عدد الصفوف المراد التعامل معها.

*** الأمر بطريقة أخرى للتوضيح:**

LIMIT رقم السجل المطلوب بدء العرض منه , عدد السجلات المطلوب عرضها

- ويفيد هذا الأمر مثلا فى حالة: الرغبة فى عرض ٣ سجلات (صفوف) بـــدأ من السجل رقم ٢..

ولتنفيذ ذلك يتم كتابة الأمر التالى:

SELECT * FROM tbref **LIMIT** 1,3 ;

ويكون ناتج التنفيذ عند ذلك هو :

```
mysql> SELECT * FROM tbref LIMIT 1,3 ;
+---------+--------+----------+------------+--------+-----------+
| name    | address| p_salary | Work_add   | phone  | mobile    |
+---------+--------+----------+------------+--------+-----------+
| Ahmed   | Damitta|  151.500 | New Damitta| 998877 | 0177777777|
| mohamed | Damitta|  111.500 | New Damitta| 112233 | 0199999999|
| Aly     | Cairo  |   95.300 | Geza       | 554466 | 0188888888|
+---------+--------+----------+------------+--------+-----------+
3 rows in set (0.00 sec)

mysql>
```

لاحظ ظهور ثلاث سجلات فقط دون الباقى..

ملاحظة هامة :

ترقيم السجلات فى الجدول يبدأ مـــن الصـــفر ولـــيس مـــن الواحـــد.. لذلك عن الرغبة فى بدء العـــرض مـــن الســـجل رقـــم ٢، يجـــب أن تكـــون قيمة number1 = 1 لأن السجل الأول يأخذ القيمة الترقيم صفر والسجل الثانى يأخذ الترقيم 1 والسجل الثالث يأخذ الترقيم 2 وهكذا..

٣-٧ التعديل فى بيانات تم إضافتها أو (تحديث البيانات):

الأمــر:

<UPDATE>

الصيــغة:

```
UPDATE table name
    SET column1=expression1, column2=expression2, ....
    [WHERE condition]
```

الصيغة بالعربية للتوضيح:

UPDATE اسم الجدول المراد الإضافة فيه
SET اسم الحقل١=القيمة الجديدة, اسم الحقل٢=القيمة الجديدة, ...
[WHERE شرط إتمام عملية الإضافة والتحديث]

مثال:

عند الحاجة لتعديل بيانات سبق إضافتها ولتكن مثلاً : يراد التعــديل فــى بيانات " Khaled " وتغيير عنوانه من Aswan إلى Cairo ...

ولتنفيذ ذلك يتم كتابة الأوامر التالية:

UPDATE tbref
SET address **LIKE** “Cairo”
WHERE name **LIKE** “khaled”;

وعند التنفيذ يظهر الشكل التالى:

```
mysql> UPDATE tbref
    -> SET address LIKE "Cairo"
    -> WHERE name LIKE "khaled" ;
Query OK, 1 rows affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql>
```

لاحظ تغير القيم التى ظهرت بالسطر:

```
Rows matched: 1  Changed: 1  Warnings: 0
```

وذلك يدل على أن عدد السجلات التى تأثرت بالتحديث والتعديل هو واحد فقط.

وللتأكد من التنفيذ الفعلى يتم كتابة الأمر التالى:

SELECT * FROM tbref ;

عند ذلك يظهر الشكل التالى:

```
mysql> SELECT * FROM tbref ;
+---------+--------+---------+-------------+--------+-----------+
| name    | address| p_salary | Work_add   | phone  | mobile    |
+---------+--------+---------+-------------+--------+-----------+
| Khaled  | Cairo  | 101.200 | Aswan       | 448899 | 0166666666 |
| Ahmed   | Damitta| 151.500 | New Damitta | 998877 | 0177777777 |
| mohamed | Damitta| 111.500 | New Damitta | 112233 | 0199999999 |
| Aly     | Cairo  |  95.300 | Geza        | 554466 | 0188888888 |
| Zyad    | Tanta  | 250.300 | New Damitta | 885599 | 0195555555 |
+---------+--------+---------+-------------+--------+-----------+
5 rows in set (0.06 sec)

mysql>
```

لاحظ تغير قيمة عنوان Khaled من Aswan إلى Cairo ..

*** ملاحظة هامة :**

وفى حالة كتابة الأمر **UPDATE** دون كتابة شـرط معـين فـى الأمـر **WHERE** .. سيؤدى ذلك إلى تأثر جميع السجلات بالتعديل ..

مثال ذلك :

فى حالة الرغبة فى تغيير الحقل p_salary إلى 100.5 وذلك فـى جميـع الحقول مرة واحدة .. فسيتم كتابة الأمر بالشكل التالى :

```
UPDATE tbref
      SET p_salary = 100.5 ;
```

وعند التنفيذ يظهر الشكل التالى :

```
mysql> update tbref
    -> SET p_salary = 100.5 ;
Query OK, 5 rows affected (0.00 sec)
Rows matched: 5  Changed: 5  Warnings: 0

mysql>
```

لاحظ ظهور رقم ٥ بالسطر التالى:

```
Query OK, 5 rows affected (0.00 sec)
```

وأيضا بالسطر التالى:

```
Rows matched: 5  Changed: 5  Warnings: 0
```

مما يدل على أن عدد السجلات التى تأثرت بالتعديل تساوى ٥ سجلات..

وللتأكد من التنفيذ يتم كتابة الأمر التالى:

SELECT * FROM tbref ;

عند ذلك يظهر الشكل التالى :

```
mysql> SELECT * FROM tbref ;
+---------+--------+---------+------------+--------+-----------+
| name    | address| p_salary | Work_add   | phone  | mobile    |
+---------+--------+---------+------------+--------+-----------+
| Khaled  | Cairo  | 100.500 | Aswan      | 448899 | 0166666666 |
| Ahmed   | Damitta| 100.500 | New Damitta | 998877 | 0177777777 |
| mohamed | Damitta| 100.500 | New Damitta | 112233 | 0199999999 |
| Aly     | Cairo  | 100.500 | Geza       | 554466 | 0188888888 |
| Zyad    | Tanta  | 100.500 | New Damitta | 885599 | 0195555555 |
+---------+--------+---------+------------+--------+-----------+
5 rows in set (0.00 sec)

mysql>
```

- لاحظ التغيير فى حقل p_salary .
- وهكذا يمكن التغيير والتعديل والتحديث لجميع بيانات السجلات عند الحاجة لذلك.

* ملاحظة هامة :

ما الفرق بين الأمرين التاليين:

INSERT .. UPDATE ?

بالطبع الفرق واضح وبسيط وهو:

- الأمر INSERT:

خاص بإضافة بيانات جديدة فى سجلات جديدة لم تكن موجـودة قبـل تنفيذ هذا الأمر ..

- أما الأمر UPDATE:

فهو لتعديل البيانات الموجودة بالسجلات أصلاً والتى تم إضافتها مـن قبل ..

أى أن الأمـر UPDATE لا يضـيف سـجلات جديـدة ولكنـه يقـوم بالتعديل فيها فقط.

وذلك يمكن ملاحظته من المثال السابق.

٧-٤ عمليات الحذف المختلفة

٧-٤-١ حذف السجلات :

الأمـــر:

<DELETE FROM>

وعن طريقه يمكن حذف جميع السجلات أوحذف سجلات معينة.

الصيـــغة:

DELETE FROM table name
  [**WHERE** condition]
  [**LIMIT** number]

الصيغة بالعربية للتوضيح :

DELETE FROM اسم الجدول
  [WHERE الشرط]
  [LIMIT عدد السجلات المراد حذفها بدء من أول سجل]

*** ملاحظة يجب الانتباه إليها:**

١- عند كتابة الأمر بالشكل التالي:

DELETE FROM table name

سيتم حذف جميع السجلات التى بالجدول دون استثناء.
ويجب الحذر عند استخدام هذا الأمر فربما يحذف مئات السجلات فى لحظة واحدة.

٢- أما عند كتابة الأمر بالشكل التالي:

DELETE FROM table name **LIMIT** number

سيتم حذف عدد من السجلات مساوى لـــ number .. من بداية الجدول.

مثال١:

- **حذف سجلات معينة:**

DELETE FROM tbref
WHERE name **LIKE** "aly" ;

عند التنفيذ يظهر التالى :

```
mysql> DELETE FROM tbref
    -> WHERE name LIKE "aly";
Query OK, 1 rows affected (0.00 sec)

mysql>
```


MR

وللتأكد من التنفيذ يتم كتابة الأمر التالى:

SELECT * FROM tbref ;

عند ذلك يظهر الشكل التالى:

```
mysql> SELECT * FROM tbref ;
+---------+--------+----------+-------------+--------+------------+
| name    | address| p_salary | Work_add    | phone  | mobile     |
+---------+--------+----------+-------------+--------+------------+
| Khaled  | Cairo  |  100.500 | Aswan       | 448899 | 0166666666 |
| Ahmed   | Damitta|  100.500 | New Damitta | 998877 | 0177777777 |
| mohamed | Damitta|  100.500 | New Damitta | 112233 | 0199999999 |
| Zyad    | Tanta  |  100.500 | New Damitta | 885599 | 0195555555 |
+---------+--------+----------+-------------+--------+------------+
4 rows in set (0.00 sec)

mysql>
```

فقد تم حذف السجل الخاص بـــ Aly.

مثال٢:

- **حذف السجلات باستخدام LIMIT:**

لحذف ٣ سجلات فقط من بداية الجدول يتم تنفيذ الأمر التالى:

DELETE FROM tbref
LIMIT 3 ;

عند التنفيذ يظهر التالى:

```
mysql> DELETE FROM tbref
    -> LIMIT 3 ;
Query OK, 3 rows affected (0.06 sec)

mysql>
```

MR

- لاحظ السطر التالى:

Query OK, **3** rows affected (0.06 sec)

فالرقم ٣ الموجود به يدل على أن السجلات التى تـأثرت بعمليـة الحذف تساوى ٣ سجلات فقط.

وللتأكد من التنفيذ يتم كتابة الأمر التالى:

SELECT * FROM tbref ;

عند ذلك يظهر الشكل التالى:

```
mysql> SELECT * FROM tbref ;
+---------+--------+---------+------------+--------+------------+
| name    | address| p_salary | Work_add  | phone  | mobile     |
+---------+--------+---------+------------+--------+------------+
| Zyad    | Tanta  | 100.500 | New Damitta | 885599 | 0195555555 |
+---------+--------+---------+------------+--------+------------+
1 rows in set (0.05 sec)

mysql>
```

مثال٣:

- حذف جميع السجلات:

DELETE FROM tbref ;

عند التنفيذ يظهر التالى:

```
mysql> DELETE FROM tbref ;
Query OK, 0 rows affected (0.00 sec)

mysql>
```


MR

وللتأكد من التنفيذ يتم كتابة الأمر التالى:

SELECT * FROM tbref ;

عند ذلك يظهر الشكل التالى:

```
mysql> SELECT * FROM tbref ;
Empty set (0.00 sec)

mysql>
```

٧-٤-٢ حذف أحد حقول (أعمدة) الجداول:

الأمـر:

DROP column name

الصيـغة :

ALTER TABLE table name
DROP column name ;

راجع الأمر صفحة ١٤٢..

٧-٤-٣ حذف جـــدول :

الأمر:

DROP TABLE

الصيـغة:

DROP TABLE <table name>

الصيغة بالعربية للتوضيح:

DROP TABLE اسم الجدول المراد حذفه

مثـال:

لحذف الجدول المسمى tbref والموجود بقاعدة البيانات refaat يـتم تنفيـذ الأمر التالى:

DROP TABLE tbref ;

وعند التنفيذ يكون:

```
mysql> DROP TABLE tbref ;
Query OK, 0 rows affected (0.05 sec)

mysql>
```


MR

وللتأكد يتم تنفيذ الأمر التالى:

SHOW TABLES ;

ليظهر:

```
mysql> show tables ;
Empty set (0.00 sec)

mysql>
```

والعبارة:

```
Empty set (0.00 sec)
```

تدل على أن قاعدة البيانات خالية من الجداول.. وذلك لان قاعدة البيانـــات refaat لم تكن تحتوى إلا على جدول واحد فقط.. وقد تم حذفه

ملاحظة :

يمكن حذف الجداول فى حالة احتوائها على بيانات، فلا فرق بين حذفها فارغة أو بها بيانات فكل سواء.

٧-٤-٤ حذف قاعدة بيانات بأكملها:

الأمر:

< DROP DATABASE>

الصيغة:

DROP DATABASE <database name>

راجع الأمر صفحة ١٠٣..

وبالطبع يمكن حذف قاعدة البيانات وهى تحتوى على جداول وبيانات، أى ليس شرطاً أن تكون فارغة.

==================

الفصل الثامن

بعض العمليات الحسابية المختلفة

يتناول هذا الفصل المحتويات الآتية :

- دالات الإجماليات فى MySQL
- أمثلة على الدوال السابقة:

MR

٨ بعض العمليات الحسابية المختلفة

- عمليات التجميع واحتساب الإجماليات :

٨-١ دالات الإجماليات فى MySQL:

اسم الدالة	الوصف
AVG(column)	متوسط القيم الموجودة فى العمود المحدد ... حيث يتم استبدال الكلمة column الموجودة بين القوسين باسم العمود المطلوب الحصول على متوسط أرقامه.
COUNT(item)	إذا قمت بتحديد عمودا (باستبدال الكلمة item باسم عمود) ستعطيك هذه الدالة عدد القيم التى ليست NULL (أى ليست قيم فارغة) فى ذلك العمود. وإذا أضفت الكلمة **DISTINCT** أمام اسم العمود ستحصل على عدد القيم الفريدة فقط (أى الغير مكررة) فى ذلك العمود. إذا قمت بكتابة **COUNT(*)** ستحصل على عدد الصفوف بالجدول بغض النظر عن القيم الفارغة NULL.

MIN(column)	للحصول على أقل قيمة أو أصغر قيمة من القيم الموجودة فى العمود المحدد.
MAX(column)	للحصول على القيمة القصوى أو أكبر قيمة من القيم الموجودة فى العمود المحدد.
STD(column)	للحصول على الانحراف القياسى للقيم الموجودة فى العمود المحدد.
STDDEV(column)	مثل STD(column)
SUM(column)	مجموع القيم الموجودة فى العمود المحدد

ملاحظة:

كل الدوال السابقة يتم استخدامها مع الأمر **SELECT** .

٨-٢ أمثلة على الدوال السابقة:

*** مثال ١:**

استخرج متوسط القيم الموجودة فى العمود المسمى p_salary داخل الجدول tbref الموجود بقاعدة البيانات refaat ..

*** الحل :**

أ) بعد تشغيل MySQL يتم فتح قاعدة البيانات refaat .. كما يلى:

```
mysql> USE refaat ;
Database changed
mysql>
```

بعد ذلك يتم كتابة الأمر التالى:

SELECT AVG(p_salary) **FROM** tbref ;

وعند التنفيذ يظهر الأتى:

```
mysql> SELECT AVG(p_salary) FROM tbref ;
+---------------+
| AVG(p_salary) |
+---------------+
| 141.9600006   |
+---------------+
1 row in set (0.00 sec)

mysql>
```

لاحظ ظهور المتوسط داخل الجدول السابق.

*** مثال ٢ :**

١- احسب عدد الصفوف(السجلات) داخل الجدول tbref الموجود بقاعدة البيانات refaat ..

٢- احسب عدد السجلات الفريدة غير المكررة فى الحقل (العمود) Work_add ..

MR

٣- قم بإضافة سجل جديد للجدول tbref بشرط أن لا يتم إضافة رقم فـي العمود mobile ومن ثم قم بحساب عدد السجلات عن طريق تحديد اسم العمود mobile .

*** الحل :**

بالطبع بعد فتح قاعدة البيانات.. يتم:

١- لحساب عدد الصفوف بالجدول tbref يتم كتابة الأمر التالى:

SELECT COUNT(*) FROM tbref ;

وعند التنفيذ يظهر:

```
mysql> SELECT COUNT(*) FROM tbref ;
+----------+
| COUNT(*) |
+----------+
| 5        |
+----------+
1 row in set (0.00 sec)

mysql>
```

لاحظ ظهور عدد جميع السجلات فى الجدول السابق.

٢- لحساب عدد السجلات الفريدة فى الحقل Work_add يتم كتابة الأمر التالى:

SELECT COUNT(DISTINCT work_add) **FROM** tbref ;

وعند التنفيذ يكون :

```
mysql> SELECT COUNT(DISTINCT work_add) FROM tbref ;
+---------------------------+
| COUNT(DISTINCT work_add)  |
+---------------------------+
|                         3 |
+---------------------------+
1 row in set (0.00 sec)

mysql>
```

لاحظ ظهور عدد ٣ سجلات فقط بالرغم من احتواء الجدول على ٥ سجلات، ويرجع ذلك إلى أن أحد القيم وهى (New Damitta) متكررة ٣ مرات فقام الأمر باحتساب هذه الثلاث قيم على أنها قيمة واحدة.

- وإذا تم تنفيذ نفس الأمر على الحقل address فماذا سيكون عد السجلات الناتجة ؟

انظر إلى الجدول التالى واستنتج الإجابة !!!

```
mysql> SELECT * FROM tbref ;
+---------+--------+----------+-------------+--------+------------+
| name    | address| p_salary | Work_add    | phone  | mobile     |
+---------+--------+----------+-------------+--------+------------+
| Khaled  | Aswan  |  101.200 | Aswan       | 448899 | 0166666666 |
| Ahmed   | Damitta|  151.500 | New Damitta | 998877 | 0177777777 |
| mohamed | Damitta|  111.500 | New Damitta | 112233 | 0199999999 |
| Aly     | Cairo  |   95.300 | Geza        | 554466 | 0188888888 |
| Zyad    | Tanta  |  250.300 | New Damitta | 885599 | 0195555555 |
+---------+--------+----------+-------------+--------+------------+
5 rows in set (0.06 sec)

mysql>
```


MR

وبمجرد النظر للعمود address تجد أن القيمة Damitta مكررة مـــرتين أى أن الأمر سيقوم بحساب هذا التكرار على انه قيمة واحدة..
أى الناتج سيكون ٤ سجلات فقط..

وللتأكد يتم تنفيذ الأمر التالى:

SELECT COUNT(DISTINCT address) **FROM** tbref ;

وعند التنفيذ يكون:

```
mysql> SELECT COUNT(DISTINCT address) FROM tbref ;
+-------------------------+
| COUNT(DISTINCT address) |
+-------------------------+
|                       4 |
+-------------------------+
1 row in set (0.00 sec)

mysql>
```

وبالطبع ستكون الاستفادة كبيرة من هذا الأمر فى حالة وجود عشـــرات أو مئات أو آلاف السجلات بأحد الجداول والمطلوب عدّ القيم الفريدة فقط.

٣- قبل إضافة سجل جديد للجدول سنقوم أولا بتنفيذ أمر حساب عدد الســـجلات فى الجدول tbref باستخدام العمود mobile .

ويكون الأمر:

SELECT COUNT(mobile) **FROM** tbref ;

وعند التنفيذ يكون الناتج:

```
mysql> SELECT COUNT(mobile) FROM tbref ;
+---------------+
| COUNT(mobile) |
+---------------+
|             5 |
+---------------+
1 row in set (0.00 sec)

mysql>
```

لاحظ أن الناتج هو ٥ سجلات وذلك كما هو متوقع..

- وعند إضافة سجل جديد للجدول بشرط أن لا يتم إضافة قيمة للحقل mobile سيختلف الناتج بالتأكيد.

ولإضافة سجل جديد نستخدم الأمر:

```
mysql> INSERT INTO tbref
    -> (name , address, p_salary , work_add , phone )
    -> VALUES
    -> ("Mahmoud" , "Damitta" , 361.7 , "Tanta" , "159951");
```

لاحظ عدم ذكر اسم الحقل mobile لعدم الرغبة فى إضافة بيانات به.

وللتأكد من الإضافة يتم تنفيذ الأمر:

SELECT * FROM tbref ;

وعند التنفيذ يكون الناتج:

```
mysql> SELECT * FROM tbref ;
+---------+--------+----------+-------------+--------+------------+
| name    | address| p_salary | Work_add    | phone  | mobile     |
+---------+--------+----------+-------------+--------+------------+
| Khaled  | Aswan  |  101.200 | Aswan       | 448899 | 0166666666 |
| Mahmoud | Damitta|  361.700 | Tanta       | 159951 | NULL       |
| Ahmed   | Damitta|  151.500 | New Damitta | 998877 | 0177777777 |
| mohamed | Damitta|  111.500 | New Damitta | 112233 | 0199999999 |
| Aly     | Cairo  |   95.300 | Geza        | 554466 | 0188888888 |
| Zyad    | Tanta  |  250.300 | New Damitta | 885599 | 0195555555 |
+---------+--------+----------+-------------+--------+------------+
6 rows in set (0.06 sec)

mysql>
```

لاحظ وجود كلمة NULL فى الحقل mobile ..

ولحساب عدد السجلات الآن فى الجدول tbref باستخدام الحقل mobile يتم تنفيذ الأمر:

SELECT COUNT(mobile) **FROM** tbref ;

وعند التنفيذ يكون الناتج:

```
mysql> SELECT COUNT(mobile) FROM tbref ;
+---------------+
| COUNT(mobile) |
+---------------+
|             5 |
+---------------+
1 row in set (0.00 sec)

mysql>
```

لاحظ أن الناتج هو أيضاً ٥ سجلات حيث أن القيمة الفارغة التى يشار إليها بالكلمة NULL لم تدخل فى حساب عدد السجلات..
ولكن فى حالة تنفيذ أمر حساب عــدد الســجلات بــالأمر (*) COUNT سيكون الناتج ٦ سجلات ...

انظر التنفيذ:

```
mysql> SELECT COUNT(*) FROM tbref ;
+----------+
| COUNT(*) |
+----------+
| 6        |
+----------+
1 row in set (0.00 sec)

mysql>
```

*** مثال ٣:**

استخرج أصغر رقم بالعمود p_salary .

*** الحل:**

يتم تنفيذ الأمر:

SELECT MIN(p_salary) **FROM** tbref ;

ويكون ناتج التنفيذ:

MR

```
mysql> SELECT MIN(p_salary) FROM tbref ;
+---------------+
| MIN(p_salary) |
+---------------+
|        95.300 |
+---------------+
1 row in set (0.00 sec)

mysql>
```

---

* **مثال ٤:**

استخرج أكبر رقم بالعمود p_salary .

* الحل:

يتم تنفيذ الأمر:

SELECT MAX(p_salary) **FROM** tbref ;

ويكون ناتج التنفيذ:

```
mysql> SELECT MAX(p_salary) FROM tbref ;
+---------------+
| MAX(p_salary) |
+---------------+
|       361.700 |
+---------------+
1 row in set (0.00 sec)

mysql>
```

*** مثال ٥:**

استخرج الانحراف القياسى للقيم الموجودة بالعمود p_salary .

*** الحل :**

يتم تنفيذ الأمر:

SELECT STD(p_salary) **FROM** tbref ;

ويكون ناتج التنفيذ :

```
mysql> SELECT STD(p_salary) FROM tbref ;
+----------------+
| STD(p_salary)  |
+----------------+
|     97.3250338 |
+----------------+
1 row in set (0.00 sec)

mysql>
```

والأمر التالى STDDEV مشابه للأمر السابق..

*** مثال ٦:**

احسب مجموع عدد القيم الحقل أو العمود p_salary .

*** الحل :**

يتم تنفيذ الأمر :

SELECT SUM(p_salary) **FROM** tbref ;

ويكون ناتج التنفيذ :

```
mysql> SELECT SUM(p_salary) FROM tbref ;
+---------------+
| SUM(p_salary) |
+---------------+
|      1071.500 |
+---------------+
1 row in set (0.00 sec)

mysql>
```

وبالطبع هذه الدوال تستخدم جميعا مع الأعمدة أو الحقول الرقمية ما عدا الدالة COUNT فيمكن استخدامها مع الحقول غير الرقمية.

====================

الفصل التاسع

phpMyAdmin

Ver 2.5.3

يتناول هذا الفصل المحتويات الآتية :

- مقدمة
- شرح أجزاء من النافذة الرئيسية
- إنشاء قاعدة بيانات
- إضافة جدول جديد
- إضافة حقل جديد للجدول
- إدخال البيانات
- عرض جميع بيانات الجدول
- عرض بيانات محددة
- وظائف إضافية
- مجموعة برامج PHPTriad

٩-١ مقدمة:

الغرض منه : التعامل مع قاعدة البيانـــات MySQL مــن خـــلال نافــذة المستعرض داخل بيئة النوافذ لجعل تنفيذ الأوامر أكثر سهولة وبساطة وســرعة ودقة...

وللحصول على أحدث إصدار من phpMyAdmin يمكنك زيارة الموقــع التالى وتحميل النسخة التى تريد:

http://www.phpmyadmin.net/home_page/downloads.php

عند ذلك يمكنك النقر على phpMyAdmin-2.5.3-php.zip وذلك لتحميــل النسخة 2.5.3 ... (لاحظ أنك كلما تابعت الموقع فإن الإصدار القديمـــة تختفـــى وتظهر بدلا منها إصدارات أحدث) ..

Final version 2.5.3

Release notes

- phpMyAdmin-2.5.3-php.tar.bz2, MD5: 1a7848cdb
- phpMyAdmin-2.5.3-php.tar.gz, MD5: 2ebb3d306d
- phpMyAdmin-2.5.3-php.zip, MD5: eec4978bbe6f
- phpMyAdmin-2.5.3-php3.tar.bz2, MD5: 296ef359

أولاً : لتتمكن من تشغيل الــ phpMyAdmin يجب أولا امتلاك وتحميــل أحد البرامج التى تقوم بتحويل جهاز الكمبيوتر العادى إلى جهاز Server وذلك بالطبع دون الاتصال بالــ Internet ...

والسبب هو أن phpMyAdmin مبرمج بلغة PHP وهـــى أحـــد لغـــات البرمجة على الشبكات.. والتى تترجم أوامرها ويظهر نـــاتج الترجمـــة داخـــل المتصفح Internet Explorer أو أى متصفح أخر للـــ Internet ..

وهذه الحالة مشابهه للغة HTML ولكن لغـــة HTML يقـوم المتصـــفح بترجمة أوامرها دون الحاجة لمترجم مساعد معه .. أمـــا الــــ PHP ولغـــات البرمجة الأخرى كالــ ASP ... تحتاج مترجم لأكوادها بالإضافة إلـــى أنهـــا تحتاج لبرنامج Server يقوم بالربط بين المترجمات والمتصفح، وقواعد البيانات وأوامر اللغة ... إلى غير ذلك من الوظائف.

ومن أمثلة هذه البـــرامج (Server) ... PWS, IIS, Omni, Apache، ولكل منهم إصداراته الخاصة به، ويتم الإعداد والتهيئة داخل نظام التشـــغيل .. ويعتبر الــ Omni من أسهل تلك البرامج، أما الــ Apache فيعتبر من أشهرهم وأقواهم بالإضافة إلى أنه مفتوح المصدر ومجانى ولكنه أكثر تعقيداً فى عمليات التهيئة الخاصة به.

ويتطلب الأمر – يفضل دائما – وبالأخص مع الــ Apache تحميل مترجم PHP (وهو المفسر لأوامر تلـــك اللغـــة) وذلـــك لإمكانيـــة تشـــغيل ملفـــات الــ phpMyAdmin.(راجع أخر هذا الفصل برنامج PHPTriad)

وقبل كل هذه البرامج يجب تحميل وإعداد وتهيئة برنامج قاعـــدة البيانـــات MySQL – كما سبق شرحه –.
وبالطبع يمكن الحصول على كل هذه البرامج من مواقعها التى تحمل اسمها على الــ Internet.

ثانياً : بعد الحصول على الــ phpMyAdmin وأحد برامج الــ Server والانتهاء من عملية الإعداد للــ Server .. يتم وضع مجلـــد phpMyAdmin داخل دليل الــ Server على القرص الصلب مثلا C:\Apache وداخل الـــدليل المختص بعرض محتوياته عبر المستعرض، وهو فى حالة استخدام الApache .. C:\Apache\htdocs

فيكون دليل الــ phpMyAdmin هو:

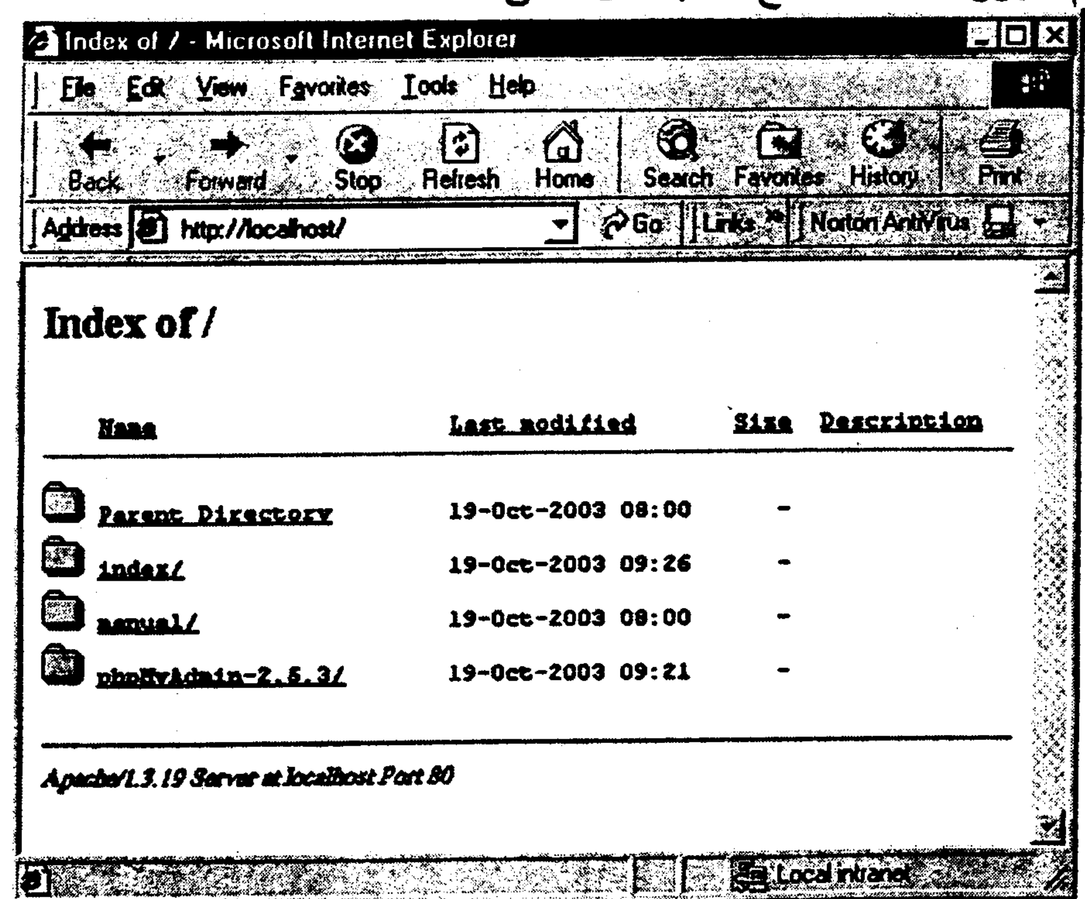
C:\Apache\htdocs\ phpMyAdmin

فمثلا الـدليل C:\Apache\htdocs هـو الخـاص بالـ Apache ، والـدليل C:\httpd\HtDocs هـو الخـاص بالـ Omni .. والـدليل C:\Inetpub\wwwroot هو الخاص بالـ IIS وهكذا ... فيتم وضع مجلـد phpMyAdmin داخل أحد هذه الأدلة إذا كان تستخدم احدهم..

ثالثاً : يتم تشغيل المتصفح وكتابة العنوان التالى :

http://localhost or http://127.0.0.1

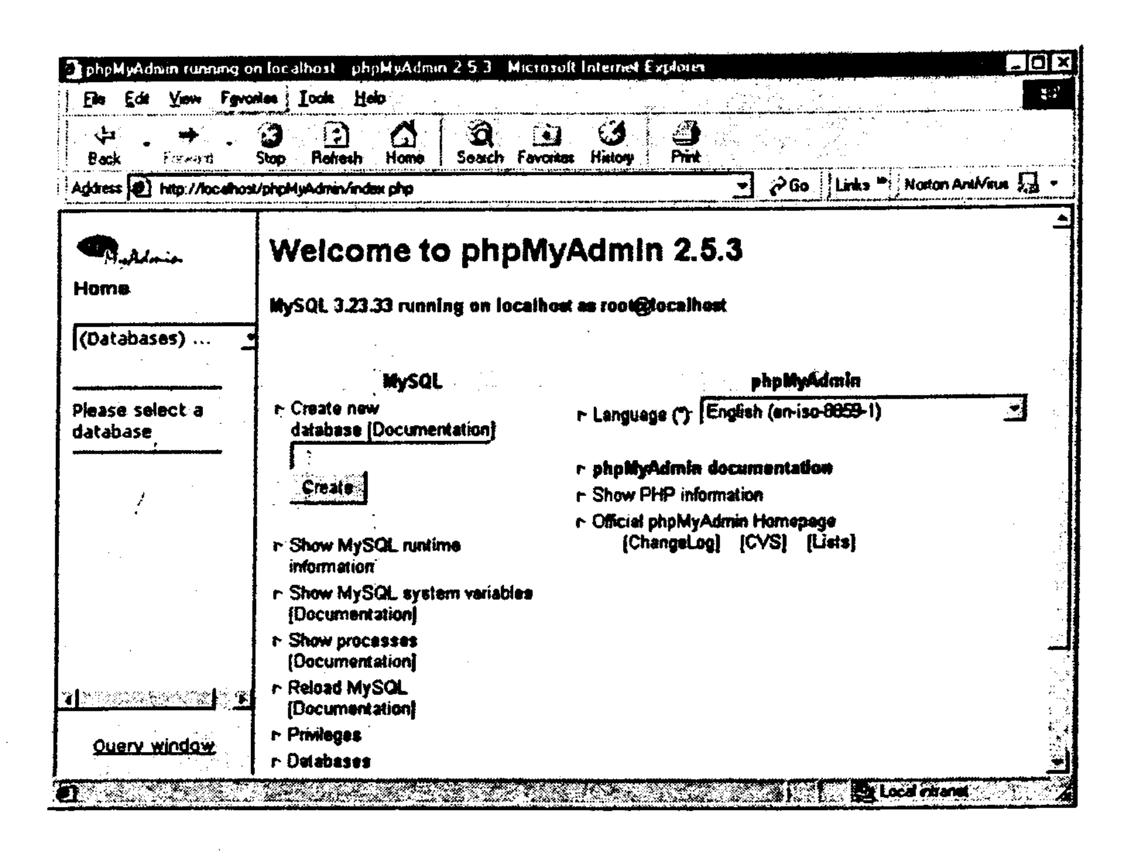
فيتم ظهور نافذة المتصفح كما بالشكل التالى:



وبالطبع هذا الشكل لن يظهر إلا بعد تشغيل أحد برامج الـ Server كما سبق.

وسنجد عدة مجلدات تم إضافة أغلبها عن طريق المستخدم، ومن بـــين هـــذه المجلدات phpMyAdmin ...

فنقوم بفتحه والنقر على الملف الموجود بداخلـــه والمســـمى index.php .. لتظهر الشاشة التالية:



الشكل السابق يوضح شاشة phpMyAdmin الرئيسية.
وكما هو ملاحظ من الشكل فهى باللغة الإنجليزية..

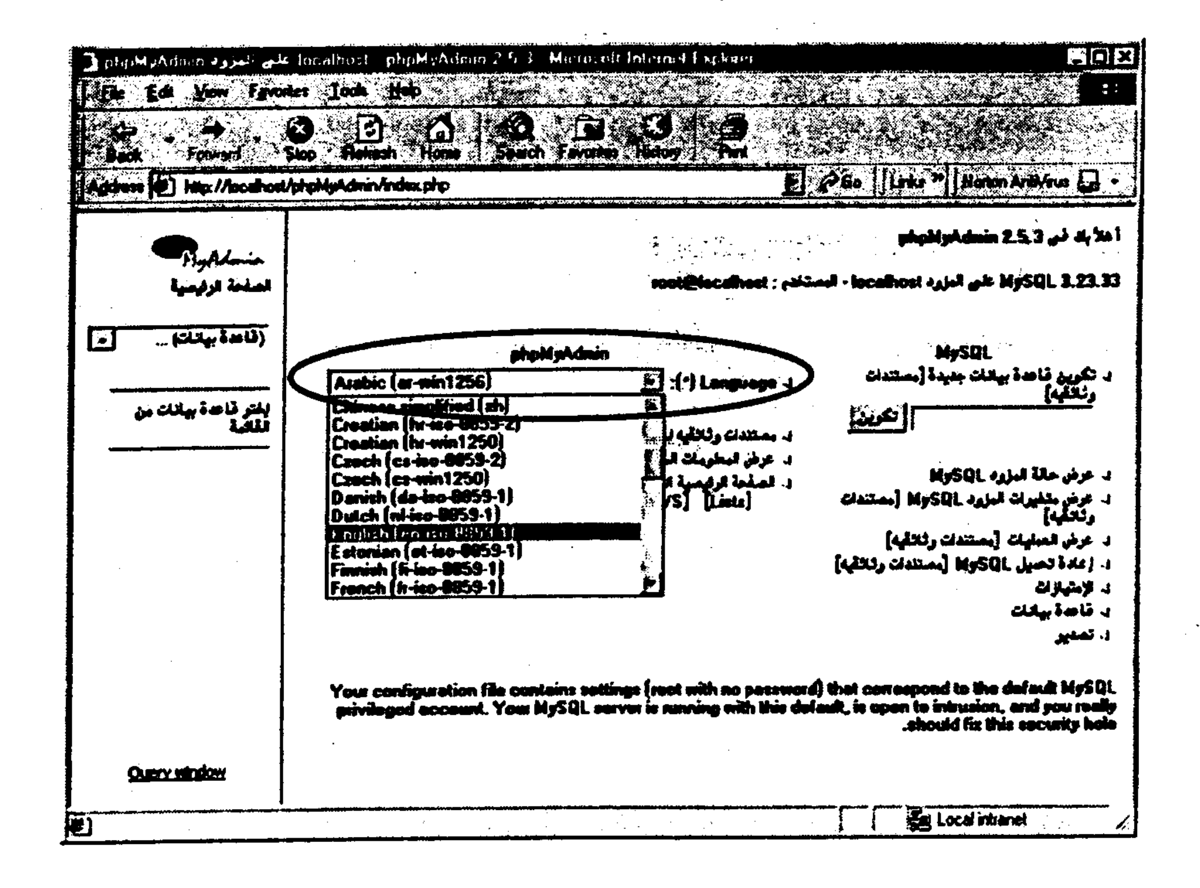
ويمكن تحويلها للعربية عن طريق النقر فوق القائمة المنسدلة المسماة Language واختيار اللغة المطلوبة ولتكن العربية Arabic ...

PhpMyAdmin

phpMyAdmin

Language (*): Arabic (ar-win1256)
Chinese simplified (zh)
Croatian (hr-iso-8859-2)
Croatian (hr-win1250)
Czech (cs-iso-8859-2)
Czech (cs-win1250)
Danish (da-iso-8859-1)
Dutch (nl-iso-8859-1)
English (en-iso-8859-1)
Estonian (et-iso-8859-1)
Finnish (fi-iso-8859-1)
French (fr-iso-8859-1)

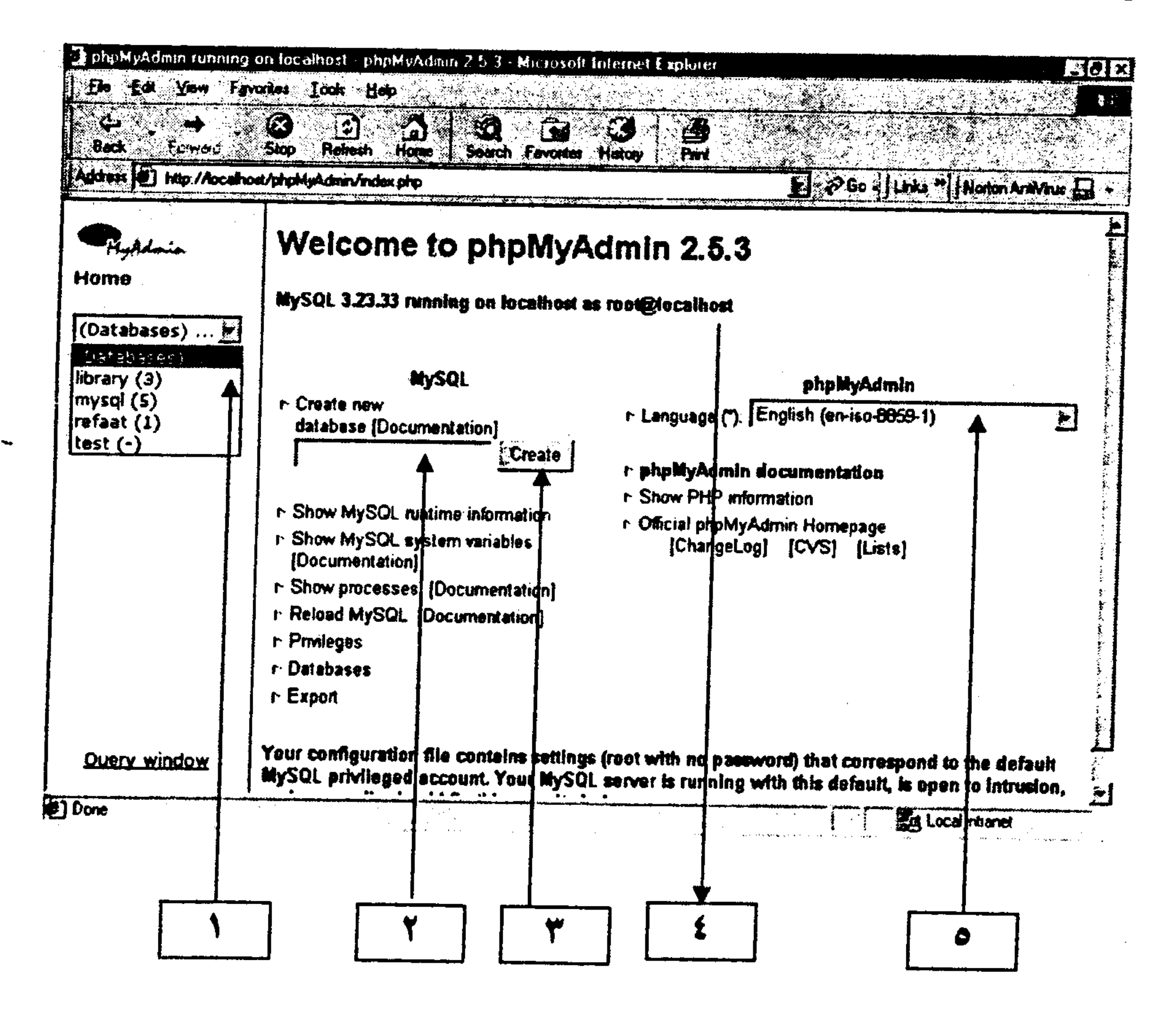
عند ذلك سيتحول الشكل السابق للشكل التالى:



وهنا سيتم الشرح باستخدام الواجهة الإنجليزية.. وذلك لاعتبارها اللغ لافتراضية للبرنامج..

٩-٢ شرح أجزاء من النافذة الرئيسية:

تتكون نافذة البرنامج من جزأين أيمن وأيسر..
الجزء الأيمن تتم به جميع العمليات.
أما الجزء الأيسر فمخصص لاختيار قاعدة البيانات المطلوب التعامل معها..
وأيضا لعرض واختيار جداول قاعدة البيانات الحالية.



MR

*** وتشير الأرقام على الرسم على الأتى:**

١- الجزء الخاص بتحديد قاعدة البيانات المراد التعامل معها .. وهذا في حالة وجود قاعدة بيانات سابقة الإنشاء يراد التعديل أو الإضافة أو الحذف منها.

٢- تشير إلى المكان الذى سيتم كتابة اسم قاعدة البيانات الجديدة فيه .. وذلك فى حالة الرغبة فى إنشاء قاعدة بيانات جديدة.

٣- الزر Create وعند النقر عليه يتم بدء إنشاء وتكوين قاعدة البيانات الذى تم كتابة اسمها فى الجزء رقم "٢".

٤- سطر يشير إلى حالة قاعدة البيانات MySQL الحالية بالإضافة إلى إصدارها.

٥- القائمة المنسدلة الخاصة بتغيير لغة واجهة phpMyAdmin إلى أى لغة أخرى.

ويمكن تنفيذ مهام أخرى من هذه النافذة ولكننا سنكتفى بعرض الخمسة أجزاء السابقة فقط.

وسيتم الآن شرح طريقة كيفية إنشاء قاعدة بيانات من خلال النافذة السابقة..

ولكن تذكر أن الأمر الخاص بذلك كان:

CREATE DATABASE اسم قاعدة البيانات الجديدة

وكان لابد من كتابته حرفيا داخل نافذة MySQL.

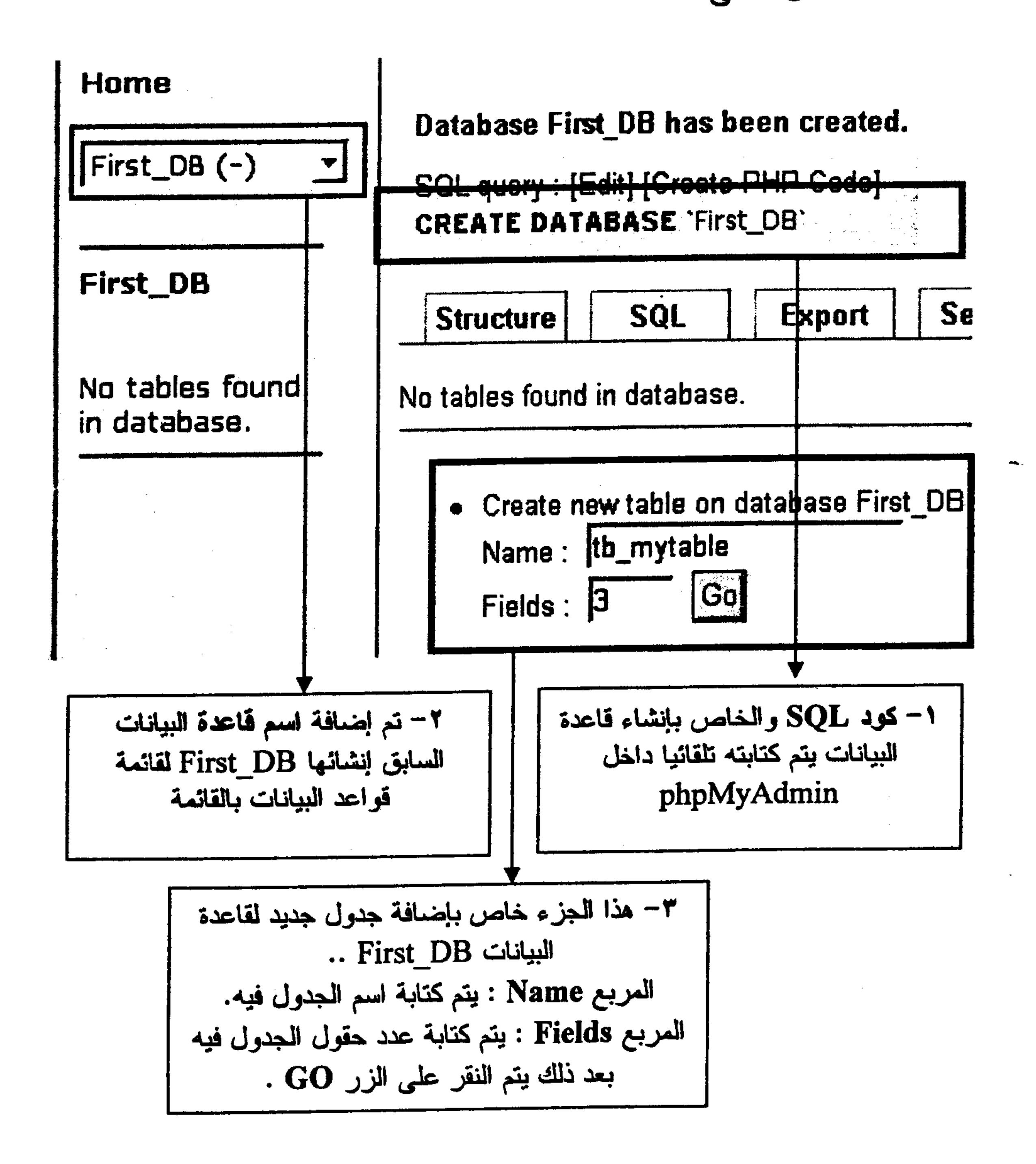
٩-٣ إنشاء قاعدة بيانات:

لإنشاء قاعدة بيانات يتم كتابة اسم قاعدة البيانات فى الجزء المسمى "Create new" (انظر الشكل السابق الجزء رقم ٢) ، ومن ثم يتم النقر على الزر Create .. وهنا سيتم كتابة الاسم First_DB لإنشاء قاعدة بيانات بهذا الاسم...

وفور النقر على الزر Create سيتم إنشاء هذه قاعدة البيانات دون كتابة أى كود.. بل إن الكود سيظهر فى أعلى الشاشة كاملا.. وأيضا سيضاف اسم هذه القاعدة فى القائمة المنسدلة بالجزء الأيسر للنافذة...

عند ذلك تظهر النافذة بالشكل التالى:

لاحظ الشكل التالى:



ملاحظة :

كتبت قاعدة البيانات بالقائمة المنسدلة كالتالى:

First_DB (-)

والعلامة (-) تدل على عدم وجود أى جدول بها وعند إضافة الجداول سيزيد هذا العدد تبعاً لعدد الجداول.

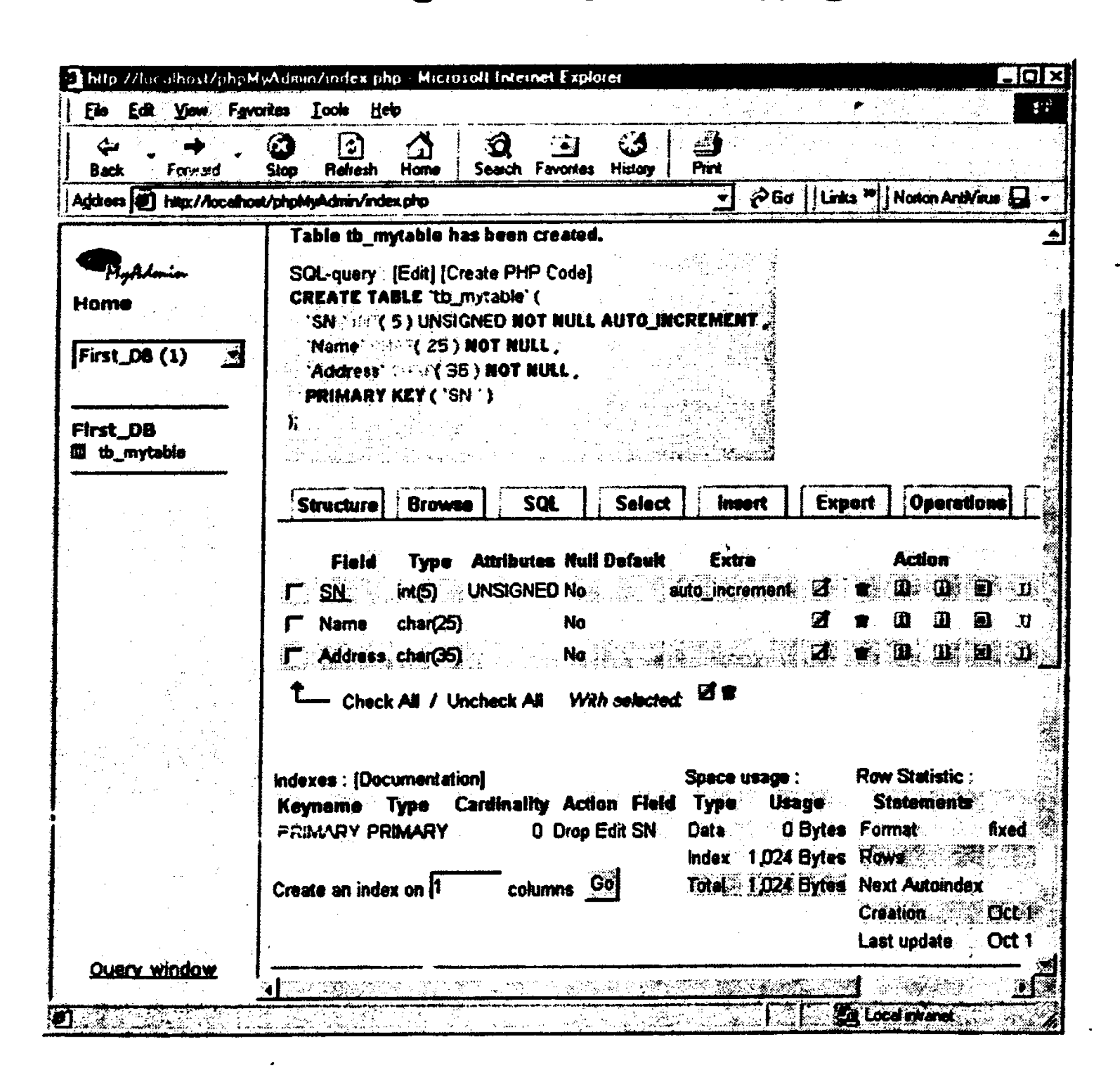
٤-٩ إضافة جدول جديد:

فى الجزء رقم ٣ فى الشكل السابق تم كتابة اسم الجدول :tb_mytable وتم تحديد عدد الحقول فيه بثلاثة حقول ..

وعند النقر على الزر GO يظهر الشكل التالى :

فيتم كتابة أسماء الحقول بالعمود الأول Field والنوع فى العمــود الثــانى Type..وبعد الانتهاء من الإضافة يكون الشكل كالتالى:

و يلاحظ سهولة التوصيف للحقول... بمجرد النقر بالفأرة علــى أى صــفة يرجى إضافتها للحقل مثل المفتاح الرئيسى وغيره..
وعند النقر على الزر Save يظهر الشكل التالى:



ومن الشكل السابق لاحظ ظهور الأتى:

١- كود الإنشاء كاملا بالـ SQL.
٢- جميع الحقول التى تم إنشائها بالإضافة إلى توصيفها.
٣- ظهور جدول خاص للفهارس، بالإضافة إلى معلومات المساحة المستهلكة.
٤- تحول (-)First_DB إلى First_DB(1) وظهور اسم الجدول أسفلها.

لاحظ:

Table tb_mytable has been created.

```
SQL-query : [Edit] [Create PHP Code]
CREATE TABLE `tb_mytable` (
  `SN ` INT( 5 ) UNSIGNED NOT NULL AUTO_INCREMENT ,
  `Name` CHAR( 25 ) NOT NULL ,
  `Address` CHAR( 35 ) NOT NULL ,
  PRIMARY KEY ( `SN ` )
);
```

والكود السابق معروض بالألوان المختلفة تمييزاً لنوع الأمر والصفة؛ والشكل التالى يوضح معلومات عن الفهارس بالجدول :

Indexes : [Documentation]

Keyname	Type	Cardinality	Action	Field
PRIMARY	PRIMARY	0	Drop Edit	SN

Create an index on 1 columns Go

Space usage :

Type	Usage
Data	0 Bytes
Index	1,024 Bytes
Total	1,024 Bytes

MR

٩-٥ إضافة حقل جديد للجدول:

من أسفل النافذة السابقة يمكن إضافة حقل جديد فى أى مكان داخل الجدول.. وعند الذهاب لأسفل النافذة نجد الشكل التالى:

- Add new field : 1 | At End of Table | Go
  - At End of Table
  - At Beginning of Table
  - After SN
  - After Name
  - After Address
- Propose table structure

Run SQL query/queries ... [Documentation]

```
SELECT * FROM `tb_mytable` WHERE 1
```

Fields: SN, Name, Address — Insert

Show this query here again

فيتم اختيار مكان الحقل الجديد ثم الضغط على الزر GO ، وعندئذ يظهـر الشكل التالى :

Database *First_DB* - Table *tb_mytable* running on *localhost*

Field	Type [Documentation]	Length/Values*	Attributes	Null	Default**
	VARCHAR			not null	

Save

وهى تشبه نافذة إنشاء الحقول..
فيتم كتابة مواصفات الحقل المطلوب ثم النقر على Save.
أنظر الشكل التالى:

Table tb_mytable has been altered.

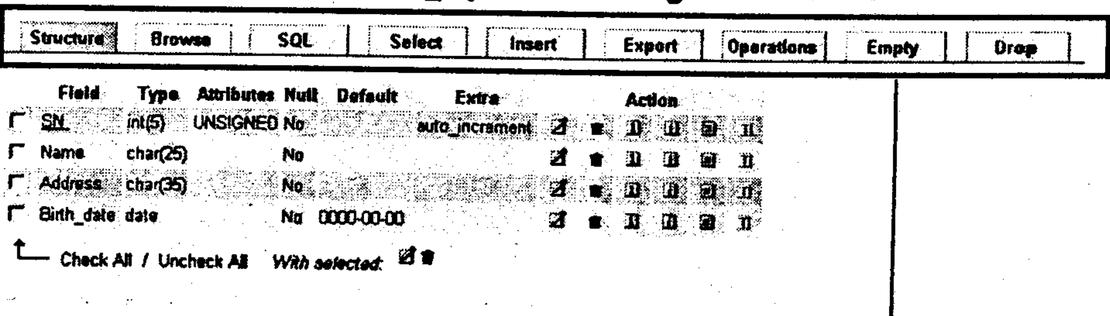
SQL-query : [Edit] [Create PHP Code]
ALTER TABLE `tb_mytable` **ADD** `Birth_date` (**NOT NULL AFTER** `Address` ;

Structure | Browse | SQL | Select | Insert | Export | Operations | Empty

	Field	Type	Attributes	Null	Default	Extra	Action
☐	SN	int(5)	UNSIGNED	No		auto_increment	
☐	Name	char(25)		No			
☐	Address	char(35)		No			
☐	Birth_date	date		No	0000-00-00		

Check All / Uncheck All *With selected:*

لا حظ ظهور الحقل Birth_date بعد الحقل Address كما تـــم تحديـــده سابقا .. وظهور كود SQL فى أعلى النافذة.



Database *First_DB* - Table *tb_mytable* running on *localhost*

Structure | Browse | SQL | Select | Insert | Export | Operations | Empty | Drop

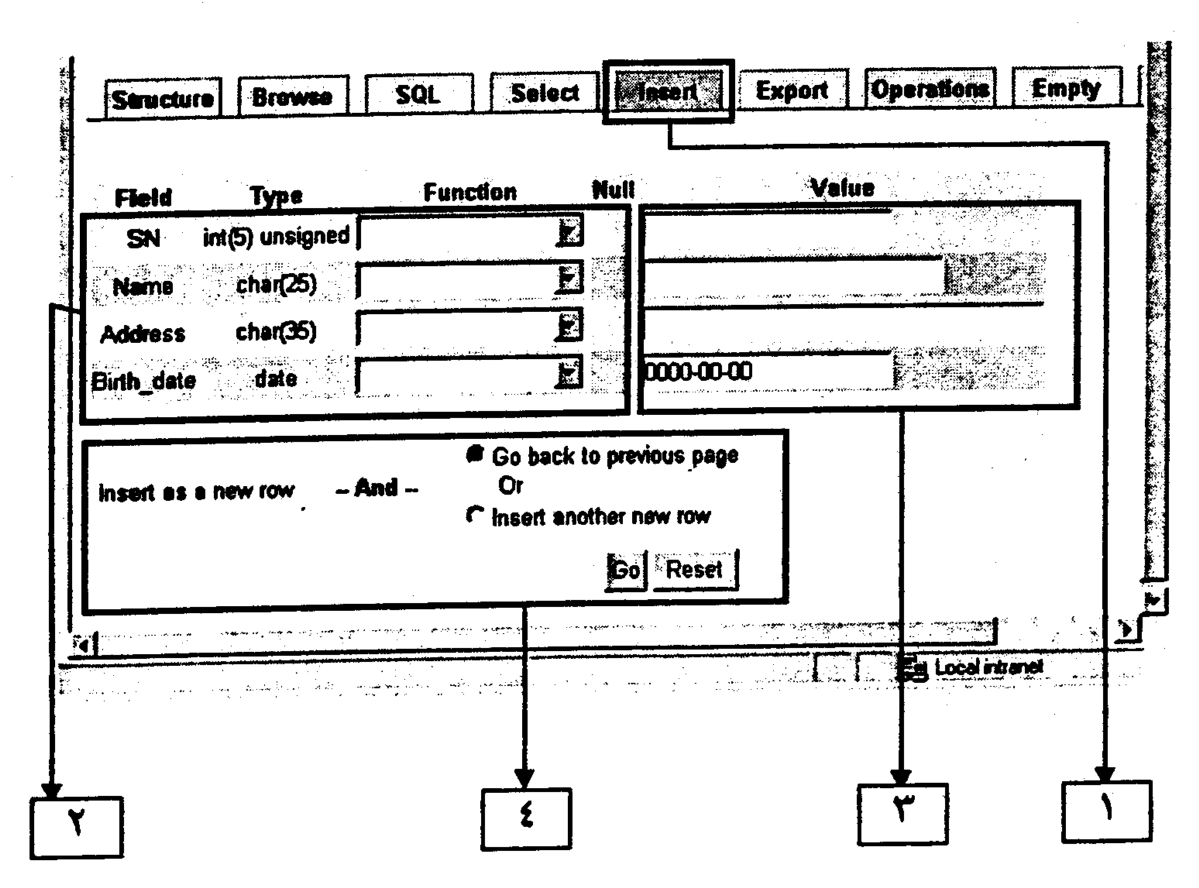
	Field	Type	Attributes	Null	Default	Extra	Action
☐	SN	int(5)	UNSIGNED	No		auto_increment	
☐	Name	char(25)		No			
☐	Address	char(35)		No			
☐	Birth_date	date		No	0000-00-00		

Check All / Uncheck All *With selected:*

عن طريق هذه الأزرار يمكن الـــتحكم بشـــكل كبيـــر فـــى الجدول .. من إضافة بيانات وعرض وتحديد وإلغاء بيانات أو إلغاء حقول ...

٩-٦ إدخال البيانات:

عند النقر على الزر **Insert** من النافذة السابقة.. يقوم البرنامج بإظهار نافذة إدخال البيانات للجدول.. كالتالى:



*** وتشير الأرقام على الرسم على الأتى:**

١- تشير إلى زر Insert أو إضافة لإضافة البيانات للجدول.
٢- تشير لأسماء الحقول وأنواعها مع إمكانية إضافة الدوال للبيانات أثناء إضافة البيانات.
٣- أماكن كتابة البيانات المطلوب إضافتها.

٤- زر التنفيذ GO أو إلغاء مسح البيانات Reset.. مع إمكانية تحديد العملية التى ستتم بعد الإضافة:

أ- العودة للشاشة السابقة (وهو الافتراضى).

ب-إضافة سجل جديد.

* يتم كتابة البيانات كما يلى:

Field	Type	Function	Null	Value
SN	int(5) unsigned			
Name	char(25)			Mohamed
Address	char(35)			Damitta
Birth_date	date			1980-10-11

* وبعد النقر على GO يتم التنفيذ والعودة للشاشة السابقة، مع إظهار الكود الخاص بالإضافة (تذكر انه الأمر INSERT INTO) كما يلى:

Inserted rows: 1
Inserted row id: 1

SQL-query : [Edit] [Create PHP Code]

```
INSERT INTO `tb_mytable` ( `SN` , `Name` , `Address` , `Birth_date` )
VALUES (
'', 'Mohamed', 'Damitta', '1980-10-11'
);
```

Structure | Browse | SQL | Select | Insert | Export | Operations | Empty

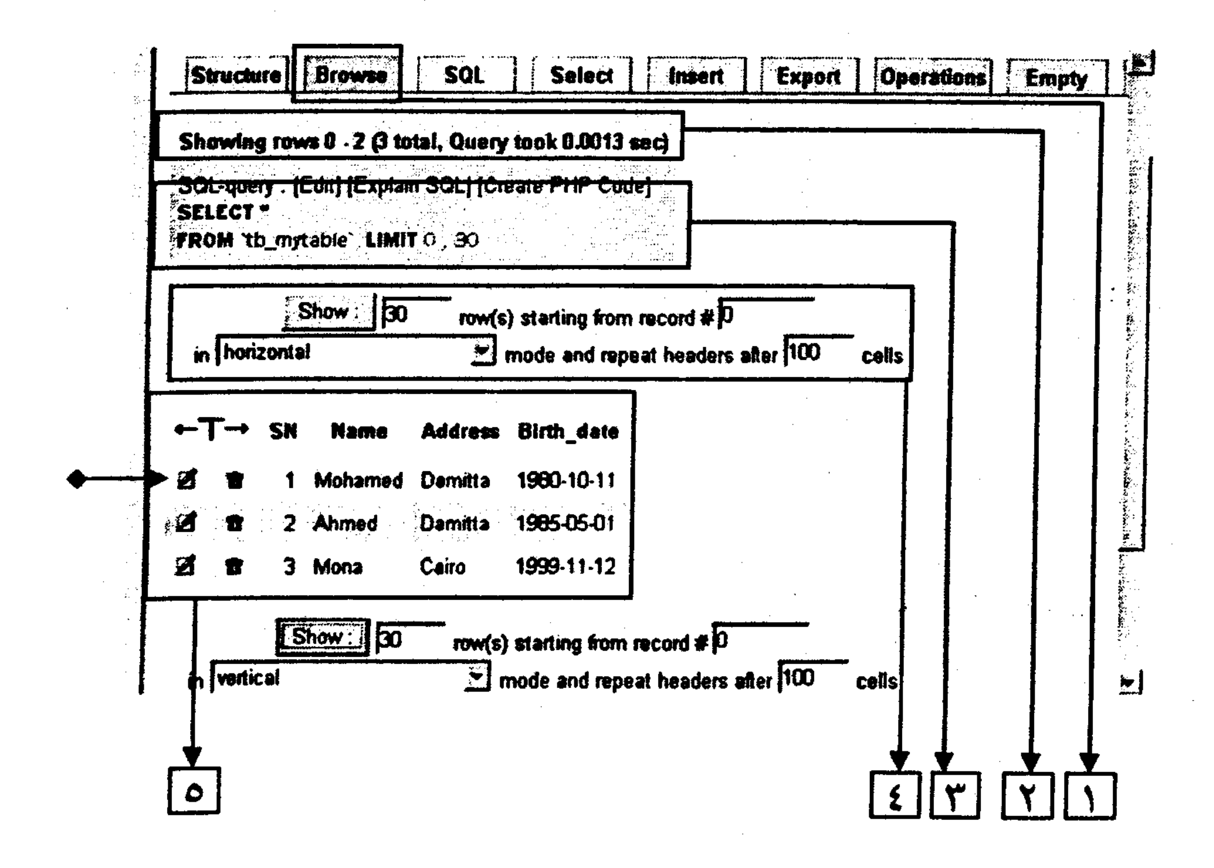
Field	Type	Attributes	Null	Default	Extra	Action
SN	int(5)	UNSIGNED	No		auto_increment	
Name	char(25)		No			
Address	char(35)		No			
Birth_date	date		No	0000-00-00		

MR

٧-٩ عرض جميع بيانات الجدول:

يمكنك عن طريق الزر المسمى **Browse** عرض جميع السجلات التى تـم إدخالها للجدول الحالى.. وكما قلنا يمكن تحديد الجداول المراد التعامل معها عن طريق اختيار اسم الجدول من الجانب الأيسر للنافذة.

وقد تم هنا إدخال ٣ سجلات للجدول tb_mytable ، وعنـد النقـر علـى الزر Browse ظهر الشكل التالى:



*** وتشير الأرقام على الرسم على الآتى:**

١- الزر الخاص بالعرض والمسمى Browse.

٢- رسالة تدل على أن العملية تمت بنجاح. وقد كان هذا السطر يظهر فى شاشة MySQL بعد تنفيذ أحد الأوامر للدلالة على أن العملية تمت بنجاح.. و إلا فكان يظهر رسالة خطأ.

٣- الكود الخاص بالعرض على هيئة أمر SQL ، وهو :

```
SELECT * FROM tbref ;
```

٤- بعض الخواص التى يمكن عن طريقها التحكم فى شكل البيانات المعروضة.. وذلك لإضافة مرونة على طريقة عرض البيانات.

مثل: تحديد عدد السجلات فى الصفحة الواحدة.

أو جعل العرض رأسى أو أفقى...

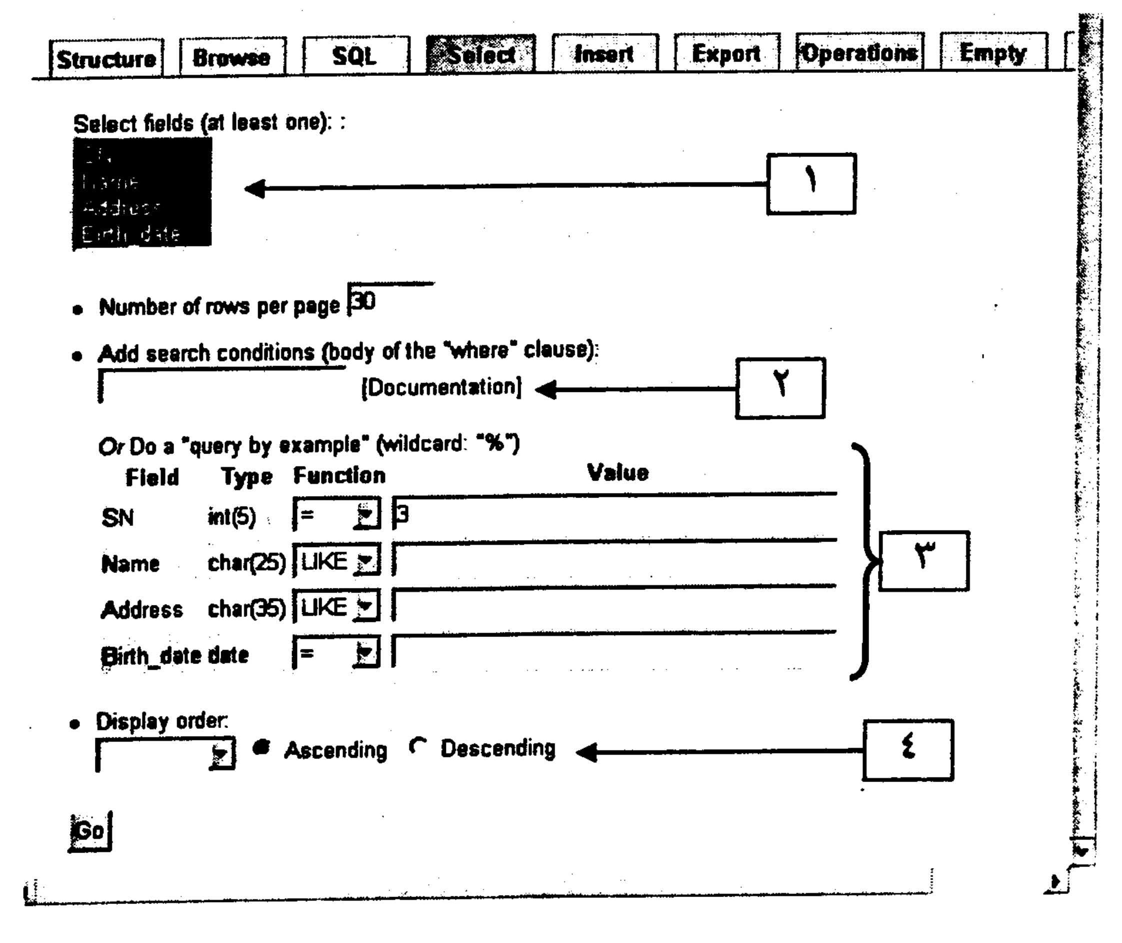
٥- وهو الجزء الهام بالشاشة..

حيث أنه يعرض السجلات التى يحتوى عليها الجدول، مع إمكانية التعامل مع كل سجل معروض.

ويمكن بالنقر على أى من الرموز الموجودة على يسار كل سجل القيام بعملية التعديل لمحتويات هذا السجل أو الإلغاء للسجل نفس ولكن فى هذه الحالة سيظهر مربع تحذير قبل تنفيذ هذا الأمر.

٨-٩ عرض بيانات محددة:

عند الرغبة فى تنفيذ أوامر العرض المختلفة والتى كانت تنفذ عن طريق الأمر SELECT فى شاشة .MySQL. يتم النقر على الزر المسمى Select لتظهر النافذة التالية:



* وتشير الأرقام على الرسم على الأتى:

١- جميع الحقول المراد إظهارها فى العرض.. ويمكن تحديد أى سجل يراد إضافته أو إلغاء التحديد بالنقر بالفأرة فوق اسمه بالزر الأيسر.

٢- لإضافة شروط لعملية العرض.. أى إضافة Where وذلك لعرض سجلات محددة مثلا.

٣- يتم إعطاء القيم المطلوب أن يتم العرض للسجلات التى تحتوى عليها.
مثل SN = 3
أو كتابة Mohamed أمام حقل الـ Name لعرض السجلات التى تحتوى على الاسم Mohamed فقط.

٤- فى حالة الرغبة فى ترتيب السجلات يتم تحديد الحقل المراد الترتيب على أساسه.. ومن ثم تحديد نوع الترتيب تصاعدى أم تنازلى..

مثال:

عند كتابة الرقم ٣ فى حقل SN ثم النقر على الزر Go فإن الناتج يكون :

phpMyAdmin
Home
First_DB (1)
First_DB
tb_mytable

Showing rows 0 - 0 (1 total, Query took 0.0013 sec)

SQL-query : [Edit] [Explain SQL] [Create PHP Code]

```
SELECT *
FROM `tb_mytable`
WHERE 1 AND `SN` = 3 LIMIT 0 , 30
```

Show : 30 row(s) starting from record # 0
in horizontal mode and repeat headers after 100 cells

←T→	SN	Name	Address	Birth_date
	3	Mona	Cairo	1999-11-12

Show : 30 row(s) starting from record # 0
in horizontal mode and repeat headers after 100 cells

Query window

Insert new row

MR

٩-٩ وظائف إضافية:

- لكتابة أوامر SQL يدوياً :

عند الرغبة فى التعامل كتابياً مع MySQL بالأوامر.. يتم الضغط على الزر SQL وذلك لفتح النافذة التالية:

MyAdmin
Home
First_DB (1)
First_DB
tb_mytable
Database *First_DB* - Table *tb_mytable* running on *localhost*
Structure | Browse | SQL | Select | Insert | Export | Operations | Empty
Run SQL query/queries on database First_DB [Documentation]
SELECT * FROM `tb_mytable` WHERE 1
Fields:
SN
Name
Address
Birth_date
Insert
Show this query here again
Or Location of the textfile :
Browse...
Go

فيمكنك كتابة ما تريد ثم النقر على الزر GO للتنفيذ.

- لتغيير اسم الجدول أو نقل الجدول لقاعدة بيانات أخرى... :

عند طريق الأمر Operations يمكنك تنفيذ كل ذلك، وهذه هى شاشه الأمر:

Home

First_DB (1)

First_DB

tb_mytable

Query window

tructure | Browse | SQL | Select | Insert | Export | Operations

- Rename table to : tb_mytable Go
- Move table to (database.table):

First_DB . tb_mytable Go

- Copy table to (database.table):

First_DB . Go

Structure only

Structure and data

Data only

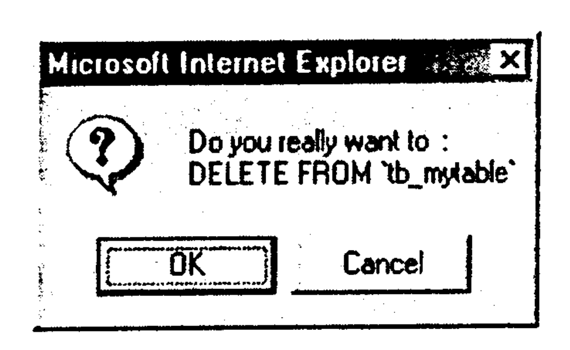
Add 'drop table'

Add AUTO_INCREMENT value

Local intranet

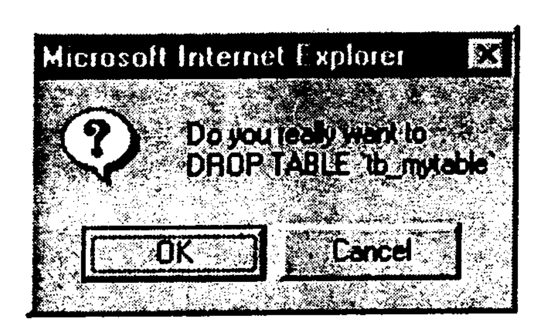
- الأمر Empty :

عند الرغبة فى حذف جميع السجلات من الجدول..
ولكن عند تنفيذه تظهر رسالة تحذيرية لذلك.
رسالة التحذير:



- الأمر Drop :

لحذف الجدول نفسه كاملا.
وأيضاً عند النقر عليه تظهر رسالة تحذيرية لذلك.
شكل الرسالة التحذيرية:



- الأمر Structure:

لعرض هيكل الجدول كاملا لجميع الحقول.

phpMyAdmin
Home
First_DB (1)
First_DB
tb_mytable
Query window

Database *First_DB* - Table *tb_mytable* running on *localhost*

Structure | Browse | SQL | Select | Insert | Export | Operations | Empty

	Field	Type	Attributes	Null	Default	Extra	Action
	SN	int(5)	UNSIGNED	No		auto_increment	
	Name	char(25)		No			
	Address	char(35)		No			
	Birth_date	date		No	0000-00-00		

Check All / Uncheck All *With selected:*

Indexes : [Documentation]

Keyname	Type	Cardinality	Action	Field
PRIMARY	PRIMARY	3	Drop Edit	SN

Create an index on 1 columns Go

Space usage

Type	Usage
Data	204 Bytes
Index	2,048 Bytes
Total	2,252 Bytes

Row Statistic

Statements	Value
Format	fixed
Rows	
Row length ø	

Local intranet

* يلاحظ من الشكل السابق أن كل حقل أمامه عدة رموز، وهذه الرموز يتم عن طريقها العديد من الوظائف وهى:
(التعديل – الإلغاء – جعل الحقل حقل رئيسى – جعله فهرس – جعله حقل فريد – جعله النص كامل)
وكل هذه الخصائص الخاصة بالحقول يمكن تنفيذها بمجرد النقر على هذه الرموز..

	Field	Type	Attributes	Null	Default	Extra	Action
☐	SN	int(5)	UNSIGNED	No		auto_increment	
☐	Name	char(25)		No			
☐	Address	char(35)		No			
☐	Birth_date	date		No	0000-00-00		

وعند النقر على الزر الأول Change يمكنك تعديل خصائص الحقل من النافذة التى ستظهر.. وهى كالآتى:

First_DB tb_mytable running on localhost - phpMyAdmin 2.5.3 - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Print

Address http://localhost/phpMyAdmin/index.php Go Links Norton AntiVirus

phpMyAdmin
Home
First_DB (1)
First_DB
tb_mytable

Database *First_DB* - Table *tb_mytable* running on *localhost*

Field	Type [Documentation]	Length/Values*	Attributes	Null	Default**
SN	INT	5	UNSIGNED	not null	

Save

* If field type is "enum" or "set", please enter the values using this format: 'a','b','c'

وبالطبع يمكنك التعديل كما تريد فى أى جزء أو صفه فى الحقل.

-----

* وبالنظر لقاعدة البيانات التى تم التعامل معها فى الجزء الأول مـن الكتـاب وهى Library نجد أن تتكون من ثلاث جداول،
انظر الشكل التالى:

library running on localhost - phpMyAdmin 2.5.3 - Microsoft Internet Explorer

Home

library (3)

library
auther
book
types

Query window

Database *library* running on *localhost*

Structure | SQL | Export | Search | Query | Drop

	Table	Action	Records	Type	Size
☐	auther		0	MyISAM	1.0 KB
☐	book		0	MyISAM	1.0 KB
☐	types		0	MyISAM	1.0 KB
	3 table(s)	Sum	0	-	3.0 KB

Check All / Uncheck All   With selected:

- Print view
- Data Dictionary
- Create new table on database library :

Name :
Fields :   Go

وبسهولة يمكن اختيار أياً منهم والتعامل معه والتعديل فيه والإضافة له وما إلـى غير ذلك.

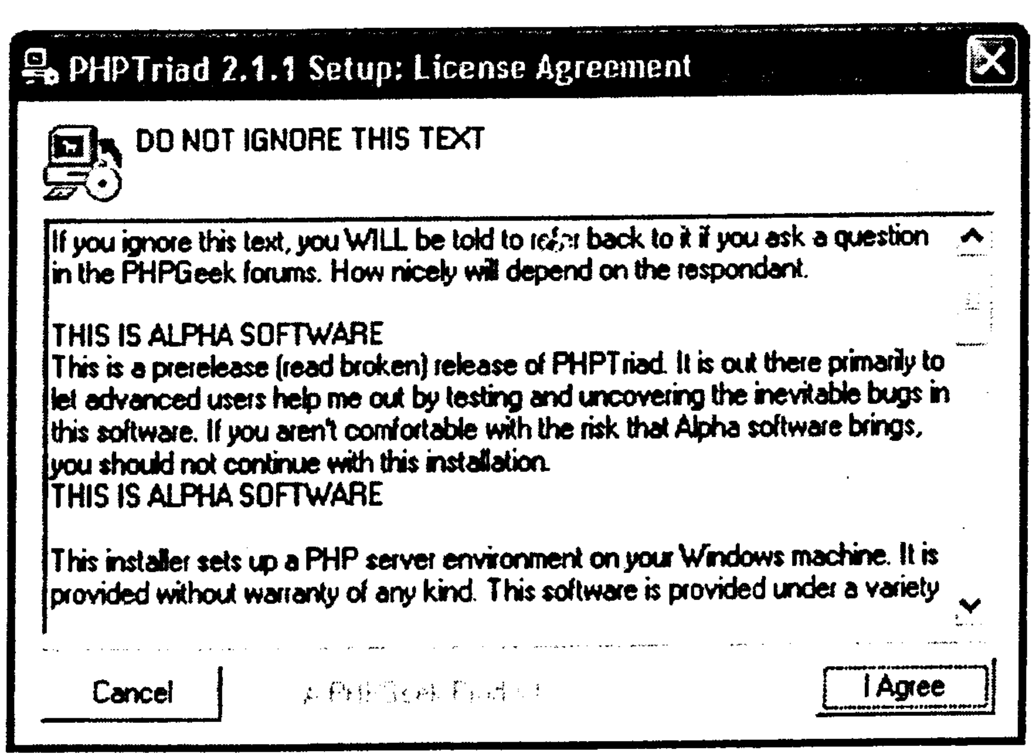
٩-١٠ مجموعة برامج *PHPTriad* :

مثال لأحد البرامج التى عن طريقها يمكنك تملّك قاعدة البيانات MySQL وبرنامج الـ Server وهو Apache بالإضافة إلى phpMyAdmin. وذلك فقط بتحميل البرنامج المسمى PHPTriad .. ويمكنك الحصول عليه من الموقع التالى:

http://sourceforge.net/projects/phptriad/

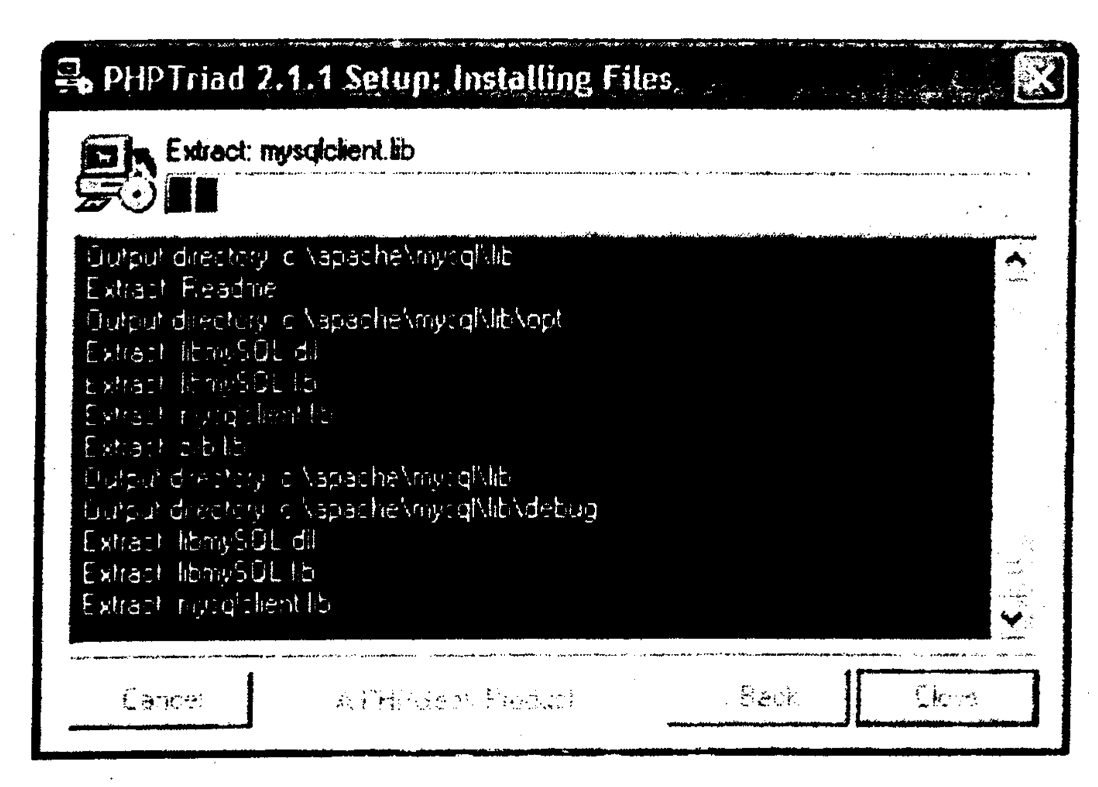
ويفضل تحميل أحدث إصدار تمت تجربته.

وبعد تحميله تتم عمليه الإعداد له على أى نوع أو إصدار من إصدارات الـ Windows أى W2k or W9x وعند ذلك يتم بدء تحميله بالنقر عليه مرتين بالفأرة ليظهر الشكل التالى :



فيتم النقر على I Agree. لمتابعة التحميل فيظهر الشكل التالى:

PhpMyAdmin



وبعد الانتهاء من التحميل يتم فتح قائمة Start لتشغيل الـ Apache عن طريق (الخطوات فى الWinXP ولن تختلف كثيرا فى Win98 أو غيره):

Start >All Programs>PHPTriad>Apache Console>Start Apache

سيظهر شاشة DOS دعها تعمل ولا تقم بإغلاقها.
وبعد التأكد من تشغيل كم من Apache والـMySQL قم بتشغيل الExplorer واكتب فى شريط العنوان:

http://localhost/phpmyadmin

لكى تبدأ التعامل مع الـ phpMyAdmin..

==============

المراجع

١- تعلم **SQL** – مهندس حيان السيد – دار الكتب للنشر والتوزيع، القاهرة ، ١٩٩٩.

٢-التطوير للويب بواسطة **PHP** و **MySQL** – لوك ويلينج ولورا طومسون- ترجمة مركز التعليم والبرمجة – الدار العربية للعلوم، لبنان، ٢٠٠٢.

3- **SAMS Teach your self MySQL in 21 Days** – Mark Maslakowski, 2000

4- www.MySQL.com

5- www.phpmyadmin.com

6- www.phpvillage.com

MR

فهرست

MR